

ARBEITSBERICHTE DES INSTITUTS FÜR MATHEMATISCHE MASCHINEN UND DATENVERARBEITUNG (INFORMATIK)

FRIEDRICH ALEXANDER UNIVERSITÄT ERLANGEN NÜRNBERG

HERAUSGEBER: H. Billing, A. Bode, W. Händler, U. Herzog,
F. Hofmann, K. Leeb, P. Mertens, H. Müller, G. Nees, H. Niemann,
H. J. Schneider, D. Seitzer, H. Wedekind

KOMPONENTEN DER DATENVERWALTUNG IN DER BÜROORGANISATION

Klaus Kratzer

DISSERTATION

BAND 19 · NUMMER 10 · ERLANGEN · DEZEMBER 1986

1987 # 13

Komponenten der Datenverwaltung in der Büroorganisation

Der Technischen Fakultät der
Universität Erlangen-Nürnberg

zur Erlangung des Grades

DOKTOR-INGENIEUR

vorgelegt von

KLAUS KRATZER

Erlangen 1986

Als Dissertation genehmigt von der
Technischen Fakultät der
Universität Erlangen-Nürnberg

Tag der Einreichung: 10.11.1986

Tag der Promotion: 17.12.1986

Dekan: Prof. Dr. H. Brand

Berichterstatter: Prof. Dr. H. Wedekind
Prof. Dr. P. Mertens

Vorwort

Die Begriffe Büroinformationssystem bzw. -automationssystem zählen seit Beginn dieses Jahrzehnts zu den gängigsten Schlagworten der informationsverarbeitenden Industrie. Die Herausforderung dieses Gebiets liegt in den völlig neuartigen Problemstellungen und Bewertungsmaßstäben, die an die Informatik-Teilgebiete Datenverwaltung, Kommunikationssysteme und Mensch-Maschine-Schnittstellen gestellt werden. Ziel meiner Arbeit in diesem Anwendungsfeld war die Aufarbeitung der dort anzutreffenden Phänomene aus der Sicht der Datenverarbeitung und der Vorstellung von Alternativkonzepten. Bei dieser Gelegenheit sollen jedoch auch die (leider nur wenigen) Vorarbeiten gewürdigt werden, die im letzten Jahrzehnt in bahnbrechenden Vorstößen Experten und das Fachpublikum für die Problemstellungen der Unterstützung und Automatisierung der Büroarbeit sensibilisiert haben. Stellvertretend für diese sei hier nur die Arbeit von Alan Kay und seiner Gruppe in den frühen siebziger Jahren erwähnt, deren Einfluß er heute mit den Worten kommentiert:

"... every good idea goes through three phases. First it is denounced as ridiculous. Next, it is said to have been obvious. Finally, its former opponents will claim to have invented it."

(KAY85, S. 280)

Ist diese Arbeit auch Produkt selbständigen Arbeitens, so ist sie dies nur im engeren Sinne und verdankt ihr Entstehen in hohem Maße der Zusammenarbeit - und oft auch der Kollision - mit Kollegen und Diskussionspartnern, die auf demselben oder

verwandten Gebieten tätig sind. Mein Dank gilt dabei in erster Linie Herrn Prof. Dr. Hartmut Wedekind, der mir bereits während meiner Studienzeit die Probleme und Lösungsansätze der Datenverwaltung nahebrachte und mich bei der Erstellung dieser Arbeit intensiv wissenschaftlich betreute und in jeder Hinsicht unterstützte. Herrn Prof. Dr. Peter Mertens möchte ich für die gute Zusammenarbeit und seine Bereitschaft zur Übernahme des Zweitgutachtens danken. Für wertvolle und ertragreiche Diskussionen bin ich den Kollegen am IMMD Erlangen, insbesondere der Forschungsgruppen VI, VII und VIII, zu Dank verpflichtet, gleichermaßen den in diesen Bereichen tätigen Studenten, die wertvolle Vorarbeiten leisteten und die kritische Auseinandersetzung mit Gegenargumenten herausforderten.

Nicht zuletzt möchte ich an dieser Stelle auch meine "externen" Diskussionspartner bei der DATEV e.G., Triumph-Adler AG, Philips Kommunikations Industrie und Siemens AG erwähnen, die mir bei der gemeinsamen Bearbeitung von Forschungsaufgaben mit industriellem Hintergrund wertvolle Informationen und Einsichten vermittelten.

Erlangen, im Oktober 1986

Klaus Kratzer

Inhaltsverzeichnis

0	EINFÜHRENDE BEMERKUNGEN	1
1	DAS BÜRO	4
1.1	DAS BÜRO ALS ORGANISATORISCHE EINHEIT	4
1.1.1	DIE DEFINITION	4
1.1.2	DIE EVOLUTION	6
1.2	DAS BÜRO ALS TECHNISCHE EINHEIT	8
1.2.1	DIE DEFINITION	8
1.2.2	DIE EVOLUTION	10
1.3	BÜROORGANISATION	13
1.4	DIE BÜROTYPEN	15
1.5	INWIEWEIT IST BÜROAUTOMATION MÖGLICH UND WÜNSCHENSWERT?	18
2	DIE DATENSTRUKTUREN	26
2.1	DIE BASISDATENTYPEN	28
2.1.1	TEXT	31
2.1.2	GRAFIK	33
2.1.3	TABELLARISCHE INFORMATION	35
2.1.4	AUDIO-VISUELLE INFORMATIONSTYPEN	39
2.2	BEARBEITUNGSWERKZEUGE (BÜROBASISPROGRAMME)	41
2.2.1	TEXTVERARBEITUNG	45
2.2.2	TABELLENKALKULATION	48
2.2.3	GRAFIKEDITOREN	50
2.2.4	KOMMUNIKATIONSWERKZEUGE	51
2.2.5	DECISION-SUPPORT-SYSTEME	55
2.3	MULTIMEDIALE DATENVERWALTUNG	57
2.3.1	KONZEPT EINER BÜROORIENTIERTEN DOKUMENTENARCHITEKTUR	60
2.3.2	NORMIERUNGSBESTREBUNGEN	73

3	DIE PRÄSENTATION	82
3.1	DER DIALOG	82
3.1.1	MENSCH-MASCHINE-INTERAKTIONEN	84
3.1.2	ZUM DIALOGBEGRIFF	93
3.2	DIE SCHREIBTISCHOBERFLÄCHE	100
3.2.1	DAS UMFELD	101
3.2.2	ALLGEMEINE ANFORDERUNGEN	104
3.2.3	DIE ARCHITEKTUR	109
3.2.4	DER ZENTRALE KATALOG	120
3.2.5	DIE ANWENDUNGSPROGRAMMSCHNITTSTELLE	126
3.2.6	ORGANISATIONSMODELLE UND INFORMATIONSÜBERTRAGUNG	130
4	BÜROPROZEDUREN	137
4.1	HANDLUNGEN IM BÜRO	139
4.1.1	ZUM HANDLUNGSBEGRIFF	139
4.1.2	HANDLUNGSVORSCHRIFTEN	143
4.1.2.1	AFINALE HANDLUNGSVORSCHRIFTEN	144
4.1.2.2	FINALE HANDLUNGSVORSCHRIFTEN - DAS PROBLEMLÖSUNGSPARADIGMA	145
4.1.2.2.1	BEMERKUNGEN ZUR PRAKTISCHEN MODALITÄT DER ERREICHBARKEIT	146
4.1.2.2.2	PROBLEME UND PROBLEMLÖSEN	147
4.1.2.2.3	AUTOMATISIERUNG VS. UNTERSTÜTZUNG DES PROBLEMLÖSENS	156
4.2	FORMULIERUNG VON BÜROPROZEDUREN	157
4.2.1	KLASSIFIKATION DER FORMULIERUNGSMECHANISMEN	157
4.2.2	PROZEDURALE MODELLE	161
4.2.3	ZIELORIENTIERTE MODELLE	168
4.3	TRANSAKTIONEN IM BÜRO	190
4.3.1	DER TRANSAKTIONS-BEGRIFF	190
4.3.2	NEUBEGRÜNDUNG DES ACID-KONZEPTES	193

5	STAND DER TECHNIK UND ZUKÜNFTIGE ENTWICKLUNGEN . .	200
5.1	ÜBERBLICK ÜBER DERZEIT VERFÜGBARE BÜROSYSTEME .	200
5.2	STAND DER DATENBANKTECHNIK	204
5.3	STAND DER KOMMUNIKATIONSTECHNIK	209
5.4	ZUR ROLLE DER KÜNSTLICHEN INTELLIGENZ IN DER BÜROAUTOMATION	211
5.5	DIE ZUKUNFT	212

ANHANG A LITERATURVERZEICHNIS

ANHANG B STICHWORTVERZEICHNIS

ABBILDUNGEN

1-1	Die Instanzen eines Büroinformationssystems . . .	10
1-2	Die Informationspyramide	13
2-1	Ebenenmodell des Datentyps	29
2-2	Idealisierte Abbildung in einem Edierwerkzeug . .	43
2-3	Beispiel für Logische Struktur und Externe Repräsentation eines Dokuments	63
2-4	Beispiel eines Hyperdokuments	66
2-5	Übersicht über die Architektur des Dokumentmonitors	68
2-6	DM-Katalog - Konstruktionsdiagramm	69
2-7	Die Dualität der logischen Struktur und der Layout-Struktur	74
2-8	Ausschnitt der ODA-Strukturen für eine Dokumentseite	76
2-9	Architektur eines typisierten Dokuments	77
2-10	Aufbau des Datenstroms für den Dokumentenaustausch	78
3-1	Herkömmliche Mensch-Computer-Kommunikation	86
3-2	Mensch-Computer-Kommunikation mit implizitem und verbreitertem explizitem Kommunikationskanal . . .	87
3-3	Diskursformen in der vertikalen und horizontalen Begründungsdimension	96
3-4	Klassifikationsschema für Standarddiskurse	97
3-5	Typische Situationen im Mensch-Computer-Dialog . .	98
3-6	Beispiel für die Schreibtischoberfläche eines Programmentwicklers	101
3-7	Beispiel für ein Partitioniertes Fenster	104
3-8	Der Abbildungsprozeß	110
3-9	Modularisierung in GDM	111
3-10	Datenfluß durch die Präsentationsflächen-Schicht	113
3-11	Datenfluß durch die Partitionsschicht	115
3-12	Datenfluß durch die Fensterschicht	117
3-13	Datenfluß durch die Schicht der Virtuellen Schreibtischoberfläche	118

3-14	Datenfluß durch die Schicht der Physischen Schreibtischoberfläche	119
3-15	GDM-Katalog - Konstruktionsdiagramm	121
3-16	Konstruktive Abhängigkeit von Standard-Organisationsmodellen	132
3-17	GDM-Katalog (Erweiterung) - Konstruktionsdiagramm	134
4-1	Praktische und theoretische Modalitäten	146
4-2	Klassifikationsschema für Probleme	147
4-3	Verfahren zur Operatorauswahl bei interpolativen Problemen	151
4-4	Weiterverarbeitung bei Mißerfolg	154
4-5	Vorwärts- und Rückwärtsverkettung	155
4-6	Ablauf der Problemlösungsunterstützung	157
4-7	Beispiel für ein Information Control Net	166
4-8	Beispiel der Ablaufdarstellung mit PrT-Netzen . .	167
4-9	Charakteristika der Büroarbeit nach dem Problemlösungsansatz	169
4-10	Konstitution des Begriffs Bürosemantik	171
4-11	Ablaufüberwachung in Objektorientierten Modellen	187
4-12	Konstitution einer Transaktion aus elementaren DB-Operationen	191
5-1	Architekturvorschläge für Nichtstandard-Datenbanksysteme	207
5-2	Schichtenarchitektur eines Kommunikationssystems .	209
5-3	Veränderung der Organisationsstruktur	214



0 Einführende Bemerkungen

Die Forschungs- und Entwicklungsaktivitäten auf dem Gebiet der Datenverwaltung haben sich im Laufe der siebziger Jahre bis heute in zunehmendem Maße in die früheren Randbereiche der Datenbank-Anwendungen verlagert. Gerade der Erfolg des Einsatzes moderner Informationstechnologien in Grafik, Animation und Telekommunikation erschloß einen Markt von bis heute kaum abschätzbarem Investitionsvolumen. Die primären Einsatzgebiete dabei sind

- Dokumentation jedweder Art
- Computerunterstütztes Design (CAD) sowie Fertigung (CAM) und Qualitätssicherung (CAQ)
- Büroaktivitäten.

Während in den ersten beiden Gebieten bereits grundlegende Normierungen und Begriffsfestsetzungen durch autorisierte Institutionen bzw. De-facto-Standards eingeführt wurden, herrscht auf dem Gebiet der Büroautomation ein kaum durch Herstellerinitiativen gezügeltes Chaos begrifflicher sowie technischer Natur. Einzig auf dem Gebiet der Kommunikationstechnik ist bereits durch das ISO-OSI-Referenzmodell (TANENBAUM81, S. 15ff) ein Grundgerüst zur Klassifikation von

Diensten und Datenstrukturen gelegt worden; die Beschreibung von anwendungsrelevanten Objekten (Dokumenten) steckt noch in den Kinderschuhen, nicht zuletzt durch die Trägheit der (halb-)staatlichen Institutionen im Bereich der Medientechnik bedingt.

Die folgenden Abschnitte orientieren sich an der Idee, daß ein Anwendungsgebiet begrifflich rekonstruiert in einem Schema erfaßt werden kann, das dann als Operationsbasis für ein Datenbanksystem dient (zu diesem Gedanken vgl. WEDEKIND80 bzw. WEDEKIND81). Bei der Übertragung dieser Idee in die Bürowelt muß jedoch das klassische *Datenbanksystem* einem *Bürodatenbanksystem* weichen, das über erweiterte Datenstrukturierungsmöglichkeiten und erweiterte Operationsmöglichkeiten verfügt und somit das klassische *Anwendungsprogramm* obsolet macht. Nichtsdestoweniger soll das Grundprinzip der Datenhaltung mit Hilfe von Datenbanksystemen, die Organisation der Daten nach einem global gültigen Ordnungsprinzip, erhalten bleiben. Der Kommunikationsaspekt wird dabei bewußt vermieden, da er, obwohl oft als das Essentielle der Büroautomation angesehen, die Abläufe im Büro nur indirekt durch bestmögliche Lokalitätstransparenz unterstützt: Es ist jedoch evident, daß im Umfeld Büro die Kommunikationstechnik einen beträchtlichen Reifevorsprung besitzt.

Im folgenden wird in Abschnitt 1 das Büro selbst dargestellt, um dem Leser einen Begriff des Anwendungsgebiets zu geben. Abschnitt 2 behandelt die relevanten Datentypen, die insbesondere nach dem Vordringen der Telekommunikation eher im unformatierten, d.h. in hohem Maße interpretativ zu verarbeitenden Bereich zu suchen sind; die dabei anzustellenden Betrachtungen führen auf den Dokumentbegriff hin, der die zentralen Manipulationseinheiten in der Bürowelt beschreibt. Abschnitt 3 betrachtet die Interaktionsmöglichkeiten, die durch ein Büroinformationssystem vermittelt werden und stellt einen Architekturvorschlag für die Gestaltung der Mensch-Maschine-Schnittstelle vor. Ziel des Abschnittes 4 ist die Präzisierung des Begriffs *Büroprozedur* im

Lichte der Anwendungsumgebung - hier werden von neuem die klassischen Konzepte der *Transaktion*, des *Triggers* und der *Integritätsbedingung* rekonstruiert werden. Zum Abschluß zeigt Abschnitt 5 die zukünftigen Linien der Weiterentwicklung auf diesen Gebieten auf.

1 Das Büro

"Der Geschichte vom Turmbau zu Babel liegt die Einsicht zugrunde, daß eine funktionierende Informationstechnik Voraussetzung einer funktionierenden Gesellschaft ist."

(Karl Steinbuch in STEINBUCH68, S. 101)

1.1 DAS BÜRO ALS ORGANISATORISCHE EINHEIT

1.1.1 DIE DEFINITION

Die Verwendung des Terminus Büro war im Zuge der Zeit einem grundlegenden Wandel unterworfen. Während in früheren Zeiten Büroarbeit als privilegierte Tätigkeit (gegenüber der reinen Fabrikarbeit) angesehen wurde, ist sie im Dienstleistungszeitalter eher als Normalfall anzusehen. So wurde denn auch der Begriff Büro von einem Statussymbol (bis ca. 1920) zu der nüchternen (geografischen) Bezeichnung eines Arbeitsplatzes (bis

ca. 1970) und schließlich zum Oberbegriff eines Bündels von Funktionen, das jeder Büroschaffende (um den Begriff *Werkstätiger* zu umgehen) auszufüllen hat. Ähnlich zahlreich und teilweise widersprüchlich-nichtssagend sind die Definitionen dieses Begriffs in der Literatur:

- "... an office will be regarded as any place, where managerial, professional, and clerical workers are engaged primarily in handling business information." (BARCOMB81, S. 2)
- "... certain basic functions are common to all (offices). In particular, they process documents." (HIRSCHHEIM85, S. 3)
- "... Zentrum des Umgangs mit der Information ..." (KARCHER82)
- und - interessant als ziemlich polemische negative Beschreibung - "'Office' is already an inadequate label for where, when and how the information society is conducted. The interpermeability ... of institutional, communication, and spatial worlds affords us revolutionary opportunities." (ELLIS82a)

Die in dieser Arbeit weiterverfolgte Definition wird im folgenden angeführt; dabei wird bewußt in Kauf genommen, daß in Einzelfällen ihre Gültigkeit in Frage gestellt werden kann.

Dem Büro als organisatorischer Einheit ist innerhalb der Gesamtorganisation ein klar von anderen Einheiten abgegrenzter Problembereich im Rahmen der Aktivitäten der Gesamtorganisation zugeordnet; Büros erzeugen, transformieren bzw. lagern Dokumente und damit Information, die auf entweder explizit oder implizit durch allgemein beachtete Verhaltensmaßregeln bestimmten Wegen das Büro erreicht oder verläßt. Information ist dabei Medium für

sprachliche Handlungen, die an anderer Stelle in nichtsprachliche Handlungen umgesetzt werden (vgl. auch AUSTIN62).

1.1.2 DIE EVOLUTION

Man kann grob drei Phasen der Entwicklung der Büroorganisation einteilen; neben organisatorischen Notwendigkeiten, d.h. der immer komplexer werdenden Aufgabe der Koordination einer großen Anzahl in verschiedener Hinsicht begabter Individuen, ist hier insbesondere auch der soziale Hintergrund des jeweiligen Zeitalters zu beachten.

Das vorindustrielle Zeitalter (bis ca. 1800). Büroaktivitäten waren anfangs Modellbildungen für Prozesse der realen Welt; durch Abstraktion von realen Handlungen oder Gegenständen waren Geschäfts- oder Verwaltungsvorgänge leichter zu erfassen, zu transportieren und auszuwerten. Man darf dabei nicht übersehen, daß die Geschichte des Büros ohne Normen, insbesondere öffentlich-rechtlicher Art, (z.B. Steuergesetzgebung, Buchführungsvorschriften) heute kaum denkbar wäre. Ein Charakteristikum des Büros war - und ist - die zentrale Datenhaltung (Registratur).

Bürotätigkeiten wurden individuell, d.h. ohne fest vorgegebenes Schema, und ausschließlich auf Papier ausgeführt: Dies brachte im Zeitalter des Analphabetismus eine ausgeprägte soziale Kluft zwischen Schreibunkundigen in der Stellung von Tagelöhnern und Wanderarbeitern einerseits und Schreibkundigen in Vertrauensstellungen andererseits hervor, ein Phänomen, das sich in den Ländern der Dritten Welt noch heute beobachten läßt.

Das industrielle Zeitalter (ca. 1800 - ca. 1950). Durch die Kapazitätsballung in Fabriken und Großmanufakturen wurde der Archetypus des vielseitigen Büroangestellten in den des spezialisierten Sachbearbeiters umgeprägt. Büroarbeiten wurden in erhöhtem Maße zur Routinesache, Arbeitsabläufe wurden durch organisatorische Maßnahmen (Formulare, Dienstweg) reglementiert, Entscheidungsprozesse wurden in immer höhere Ebenen der Organisation verlagert.

Das Zeitalter der Information (ca. 1950 - Gegenwart). Dieser Zeitraum ist durch einen lawinenartigen Anstieg der zu bewältigenden Informationsflut geprägt. Es ist interessant, daß die technischen Hilfsmittel zu dieser Bewältigung neue Informationen in diesen Kreislauf abgeben und so gleichzeitig zu einer weiteren Verschärfung dieser Entwicklung beitragen. Als Gegenmaßnahme findet man dabei die Diversifikation der Präsentation, die es den Entscheidungsträgern ermöglicht, die Überlast an Daten durch geeignete Verdichtung abzubauen und doch die für ihre Tätigkeit entscheidenden Trends abzulesen. Die Fähigkeit oder Unfähigkeit zum Umgang mit informationsverarbeitenden und -übertragenden Geräten, teilweise durch das amerikanische Schlagwort *Computer (Il-)literacy* bezeichnet, schafft neue soziale Klüfte analog zur Zeit des Analphabetismus.

1.2 DAS BÜRO ALS TECHNISCHE EINHEIT

1.2.1 DIE DEFINITION

Der Bürobegriff, der das Fachgebiet Büroautomation in der Informatik geprägt hat, ist durch seinen Gebrauch in leider meist nicht sehr seriösen Fachmedien geprägt. Nachdem das Marktsegment als überaus profitabel erkannt worden war, wurde viel neuer, aber unausgegorener, jedoch auch viel alter Wein in neue Schläuche abgefüllt; es entstand eine Vielzahl von Insellösungen, wobei im Prinzip irrelevante Attribute eines Systems wie Maus in puncto Hardware oder Kalenderverwaltung auf der Softwareseite sofort einen schlagenden Beweis für Bürotauglichkeit darstellen sollten. Die Tätigkeiten im Büro an sich und die Anwendbarkeit eines Werkzeugs in einer gegebenen Situation ist dabei noch völlig außer acht gelassen worden.

Die technische Bestimmung des Bürobegriffs geht einher mit der Festlegung des Begriffs Büroautomation. Eine Definition, die auch im folgenden maßgeblich sein wird, gibt Barcomb:

"Office automation is the incorporation of appropriate technology to help people manage information." (BARCOMB81, S. 1).

Streng genommen heißt dies, daß

- (i) neben der Verfügbarkeit von Technologie an sich auch deren Angemessenheit in der Anwendungssituation festgestellt werden muß und daß

- (ii) von Automatisierung wie etwa im Sinne der Fertigungsautomatisierung im allgemeinen nicht die Rede sein kann; als hinreichend vornehmes Ziel ist dagegen die Unterstützung der Bürotätigkeit anzusehen.

Im engen Bezug hierzu steht der Begriff *Büroinformationssystem*, der die Konkretisierung der unter Büroautomation subsumierten Absichten darstellt.

Ein *Büroinformationssystem* umfaßt die integrierte Anwendung aller im Büro zum Zwecke der Übertragung und Verarbeitung von Information relevanten Technologien; zu diesen zählen Datenverarbeitungsanlagen und ihre Peripheriegeräte, analoge und digitale Übertragungseinrichtungen und alle darauf aufbauenden Dienste (z.B. Telefon, Telefax, Teletex, Bildschirmtext), Vervielfältigungseinrichtungen (Kopierer) sowie alle zur deren effizienter Nutzung erforderlichen Programme.

Im Rahmen der vorliegenden Arbeit wird der Begriff in erster Linie in bezug auf die Datenverarbeitungsanlagen und deren Programme angewendet werden. Abb. 1-1 zeigt eine Grafik der verschiedenen Instanzen eines *Büroinformationssystems*, die gleichzeitig die Perspektiven symbolisieren, aus denen Büroarbeit betrachtet werden kann; dabei wird noch bewußt darauf verzichtet, dem Büroautomationssystem ein ganzheitliches Konzept zugrunde zu legen.

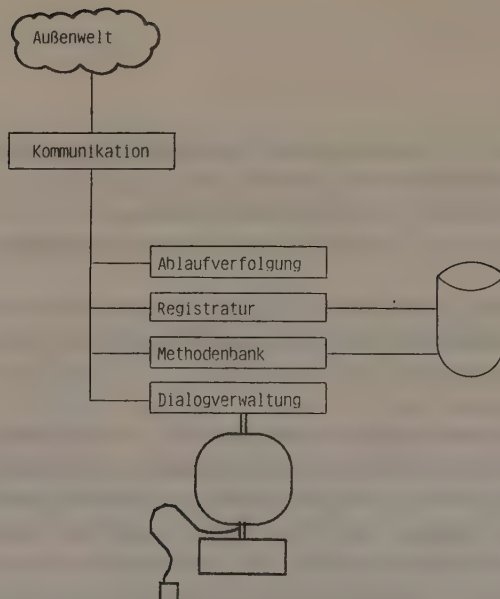


Abb. 1-1: Die Instanzen eines Büroinformationssystems

1.2.2 DIE EVOLUTION

Die Geschichte der Büroautomation ist eng gekoppelt an die technischen Möglichkeiten der jeweiligen Epoche; man kann dabei diverse Meilensteine erkennen (siehe z.B. HIRSCHHEIM85, S. 7ff):

ca. 1870 Die erste marktreife *Schreibmaschine* wird zur Basis der Massenkommunikation.

ca. 1920 Das *Telefon* wird zum beherrschenden Faktor zwischenmenschlicher Kommunikation über längere Distanzen und damit zum Vorreiter der Telekommunikation.

- ca. 1950 Die Einführung der Methoden des *Operations Research* führt zu ersten Lösungsansätzen für nichttriviale Optimierungs- und Entscheidungsprobleme und schafft so die Grundlage für spätere entscheidungsunterstützende Systeme (*Decision-Support-Systeme*). Die ersten marktreifen Fotokopiergeräte erlauben schnelle und kostengünstige Vervielfältigung von Dokumenten innerhalb einer Organisation.
- ca. 1960 Elektronische Datenverarbeitung ist aus dem wissenschaftlichen Bereich in die Geschäftswelt vorgedrungen. Erste Ansätze der *Datenfernverarbeitung* gehen einher mit der Entwicklung von Betriebssystemen für *Teilnehmer-* bzw. *Teilhaberbetrieb*. *Dialogprogramme* lösen schrittweise die althergebrachte Stapelverarbeitung ab. EDV dringt an die Arbeitsplätze von Sachbearbeitern vor, die in einem gut strukturierten Arbeitsgebiet eine begrenzte Menge von Aktionen ausführen, die sich wiederum auf *Dialogtransaktionen* abbilden lassen.
- ca. 1970 *Digitale Netzwerke* erlauben die Verteilung von Aufgaben und, unabhängig von geografischen Gegebenheiten, den Daten-, Funktions- und sogar Lastverbund unter den betroffenen Rechnerknoten. Lokale Hochgeschwindigkeitsnetzwerke führen zur Transparenz der Verarbeitung und des Zugriffs innerhalb eines Büros. Durch den Ersteinsatz von *Textverarbeitungssystemen* wird der erste Schritt zu flexiblen interaktiven Werkzeugen vollzogen.
- ca. 1980 Diese Phase ist durch durch eine krasse Umkehrung des Kostenverhältnisses zwischen Soft- und Hardware gekennzeichnet. Alle Bestrebungen laufen auf den Aufbau *lokaler Verarbeitungskapazitäten* hin. Insbesondere das Vordringen der sog. *Personal Computer* im privaten wie

wirtschaftlichen Sektor und der Erfolg neuer, dialogintensiver Anwendungsprogramme sind in diesem Zusammenhang zu nennen. Ihre professionellen Pendanten bringen die Verarbeitungskapazität eines herkömmlichen Minirechners im Einplatzbetrieb an den Arbeitsplatz (Arbeitsplatzrechner bzw. Workstations), wobei als Pilotanwendungen hierfür die Konstruktionsarbeitsplätze im CAD/CAM-Bereich zu nennen sind. Die in diesem Zusammenhang bahnbrechende Entwicklung war zweifelsohne das System 8010 der Xerox Office Systems Division, das auch unter der Bezeichnung Star bekannt wurde. Seine extrem hohe Verarbeitungskapazität in Verbindung mit neuartigen Konzepten der Kommunikation (Ethernet), Benutzerführung (Simulation einer Schreib-tischoberfläche) und Systemsoftwarearchitektur (Objektorientiertes Programmiersystem) hatte maßgeblichen Einfluß auf die Gestaltung von Systemen dieses Anwendungsgebietes (vgl. hierzu auch CURRY83); indes, der Markterfolg ist ihm nicht zuletzt aufgrund der fehlenden Integrationsmöglichkeit mit funktional supplementären Systemen versagt geblieben. Die derzeit gängigen Systeme dieser Klasse suchen allsamt die Anlehnung an Standardkomponenten (z.B. Standardbetriebssysteme wie MS-DOS oder UNIX), da sich die Verfügbarkeit einer großen Auswahl an integrierbaren Hard- und Softwarekomponenten als der letztlich entscheidende Faktor für ein Durchsetzen auf diesem hartumkämpften Markt erwiesen hat.

1.3 BÜROORGANISATION

Hinsichtlich der globalen Organisation von Büroarbeit kann man sich an der klassischen Dreiteilung in strategische, dispositive und operative Ebene orientieren, die in Abb. 1-2 in Form einer Informationspyramide wiedergegeben ist und nach WALBRÜCKE (S. 333) einen der entscheidenden Aktionsparameter bei der Konfigurierung von Büroinformations- und -kommunikationssystemen darstellt. Mit Blickrichtung auf Büroorganisation und an den agierenden Personen orientiert, werden diese Ebenen von Naffah wie folgt beschrieben:

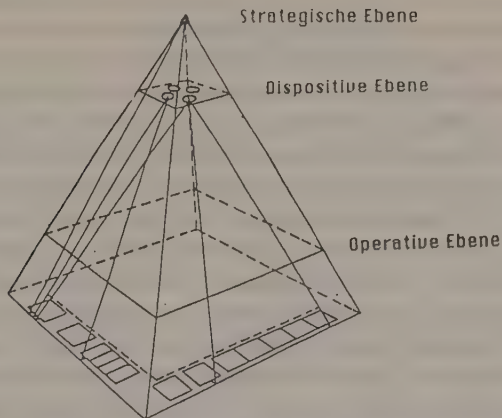


Abb. 1-2: Die Informationspyramide

"Les administrateurs sont les responsables des grands choix dans l'organisation. Ils décident des orientations, du lancement de nouveaux projets, de l'embauche, de la croissance de leurs entreprises. (Strategische Ebene; Anm. d. Verf.)

Les techniciens sont ceux qui analysent et traitent des problèmes spécifiques, en prenant des décisions techniques dans le cadre de leur profession. Ils sont les centres d'exécution et de production de l'organisation. (Dispositive Ebene; Anm. d. Verf.)

Les clercs et les secrétaires interviennent directement dans les circuits de mouvement de l'information: saisie, mise en forme, circulation, élaboration, stockage. Ils prennent des décisions à leur niveau, mais agissent principalement pour la préparation des décisions à prendre dans les autres catégories. (Operative Ebene; Anm. d. Verf.)" (NAFFAH79, S. 10 ff)

Die im vorhergehenden Abschnitt angesprochenen technischen Rationalisierungen erfassen im allgemeinen den operativen Bereich. Es wird jedoch bereits die Herausforderung sichtbar, die sich beim Übergang in den dispositiven Bereich stellt: Während im operativen Bereich gemeinhin eine strikt zu befolgende Beschreibung aller Arbeitsvorgänge vorliegt (nach KARCHER82, S. 361f: Filtern, Sortieren und Ablegen von Informationen, Be- und Verarbeitung nach standardisierbaren Mustern sowie Verteilung von Informationen), ist im dispositiven Bereich das pseudo-kreative Element vonnöten, das vage als *Fingerspitzengefühl* oder *Einfühlungsvermögen*, umgangssprachlich evtl. als *gesunder Menschenverstand*, bezeichnet werden kann. Im Abschnitt 4 dieser Arbeit wird dargestellt werden, daß dies statt der einfachen Ausführung einer Prozedur eine Konstitution einer situationsangemessenen Handlung aus Elementarhandlungen, mithin eine Metahandlung, erfordert und gleichzeitig der Mythos der Kreativität als solcher entlarvt wird. Komplexe Aktionen auf der strategischen Ebene sind im allgemeinen nicht Gegenstand einer Automatisierung; jedoch stehen für die auf den unteren Ebenen erzeugten Daten Verdichtungsverfahren sowie Präsentationsverfahren für eine Vielzahl von Medien zur Verfügung.

1.4 DIE BÜROTYPEN

Durch Abstraktion von real auftretenden Büros lassen sich *Bürotypen* bilden, die sich durch den Grad der Wohlstrukturiertheit der in ihnen anfallenden Arbeitsgänge unterscheiden. Im Zuge der Simulation von Bürovorgängen (Ermittlung des Durchsatzes, Erreichbarkeit von Zuständen) haben sich durch Typisierung des Arbeitsprofils folgende Grundtypen herauskristallisiert (vgl. LEAVENWORTH78):

- Das *Linienbüro* (*service-oriented office*) zeichnet sich durch eine sehr geringe Zahl verschiedenartiger Aufgaben und eine eindeutige Festlegung der Arbeitsfolgen aus; als Beispiele sind Flugreservierung, Autovermietung sowie im öffentlichen Bereich z.B. die Lohnsteuerstelle zu nennen. Leavenworth sieht die Aktivität eines derartigen Büros in Intensität und Art als *funktional* abhängig von durch die Außenwelt oder andere Büros erzeugten Stimuli und deshalb als individuell oder im Verbund vorhersehbar an. Für die Automatisierung bzw. Unterstützung der Abläufe bieten sich die herkömmlichen Mittel der Datenverarbeitung, das klassische Anwendungsprogramm mit Transaktionsverwaltung und Standard-Datenbanksystem, an.
- Das *Stabsbüro* (*principal support office*) wird in erster Linie durch *selbsterzeugte* Stimuli getrieben. Feste Ablaufsequenzen sind nicht bekannt, man kann jedoch ein Tätigkeitsprofil durch die zeitlichen Anteile verschiedenartiger Aktivitäten an der Gesamtaktivität bilden. Im Gegensatz zum Linienbüro erfolgt eine Unterstützung bisher weitgehend punktuell durch Einsatz dedizierter Werkzeuge für bestimmte Aktivitätsklassen (Textverarbeitung, Kalkulation etc.), wobei sowohl die Definition von Abläufen als auch die Integration ver-

schiedener Datentypen kaum möglich sind: Datenstrukturen haben keine Existenzberechtigung per se, sondern nur im Kontext des Werkzeugs, das sie erzeugt. Beim Übergang von der isolierten Betrachtung eines Einzelarbeitsplatzes zur gesamtheitlichen Betrachtung des Zusammenwirkens von Arbeitsplätzen innerhalb eines Systems ist auch die Darstellung und Behandlung der inhärenten Parallelität von Vorgängen erforderlich.

Nach DODSWELL⁸³ sind aufgrund diverser Aktivitätsanalysen die Tätigkeiten zeitlich im Durchschnitt folgendermaßen verteilt:

	FP(%)	FE(%)	SE(%)	SB(%)
Kommunikation	60	33	15	12
Dokumenterzeugung	18	23	60	40
Dokumentverwaltung	20	40	20	40
diverse	2	4	5	8

FP = Führungsperson (typisch Stabsbüro)

FE = Fachexperte (typisch Stabsbüro)

SE = Sekretariat (typisch Stabsbüro)

SB = Sachbearbeiter (typisch Linienbüro)

Beachtenswert in dieser Aufstellung ist der von links nach rechts abnehmende Anteil kommunikativer Tätigkeiten; die Automatisierung von Stabsbüros wird deshalb in weit höherem Maße Protokollkomponenten zur Kommunikation zwischen Büros berücksichtigen müssen.

Leavenworths Typisierung wird in groben Zügen durch Panko und Sprague (PANKO⁸²) bestätigt, die dabei insbesondere als Maß für die Effizienz des Linienbüros die Produktivität (Transaktionsrate) und im Fall des Stabsbüros das Gesamtverhalten innerhalb der Organisation heranziehen; die letztere Größe ist schon deshalb schwer in ein vergleichbares Maß überzuführen, da

bei diesem Typ aufgrund des höheren Kommunikationsanteils an der Gesamtaktivität das Schnittstellenverhalten eine entscheidende Rolle spielt.

Eine wesentlich tiefer gehende Unterscheidung trifft Malone in seiner Studie über menschliche Arbeitsweisen im Büro (MALONE83). Er teilt dabei die Einzelaktivitäten in folgende Klassen auf und orientiert sich dabei an der Verteilung arbeitswichtiger Dokumente einerseits und deren Fluß bzw. den Verfahren zu ihrer Auffindung andererseits:

- *Finding.* Bei einer überwiegend durch Routinearbeiten geprägten Arbeitsumgebung (typisch: Sachbearbeiter) wird die Dokumentenverteilung strikt organisiert sein; der Dokumentfluß und die Ablage sind peinlich genau festgelegt. Die dabei auftretende Organisationsform wird durch den *Aktenordner (file)* symbolisiert, der die Auffindung gezielt durch Deskriptoren ermöglicht. Problemstellungen in dieser Organisation sind insbesondere die automatische Deskribierung von Dokumenten sowie die Beschleunigung ihrer Wiedergewinnung aufgrund dieser Deskriptoren. Ein derartiges Büro wird intuitiv als "ordentlich" bezeichnet.
- *Reminding.* Im Gegensatz dazu wird im Büro einer Führungsperson oder eines Fachexperten eine "chaotische" Organisation vorherrschen. Man verzichtet auf eine streng festgelegte Prozedur zur Dokumentenauffindung, nicht zuletzt, weil aufgrund der Vielfalt der abzulegenden Informationen ein vernünftiges Deskribierungskriterium nicht antizipiert werden kann und sowieso die physische Ablage nur nach einem Deskriptor vorgenommen werden könnte. Vielmehr erfolgt eine äußerst grobe Unterteilung in Kategorien und der Aufbau von *Stapeln (pile)*, die einerseits bei der Wiederauffindung durch ihre physische Anordnung ein grobes Maß

für die zeitliche Struktur ihres Eintreffens bilden, jedoch auch, und wichtiger, eine *Erinnerungsfunktion* an wichtige Termine, dringende Bearbeitungsschritte u.ä. haben.

Die Funktionen *finding* und *reminding* können grob an das Linien- bzw. Stabsbüro assoziiert werden. Man wird jedoch, wie auch in Feldstudien gezeigt wurde (vgl. u.a. PANKO84), kaum einen dieser Grundtypen in der Realität vorfinden. Aus diesem Grunde sind die Bedürfnisse beider Typen für eine Betrachtung der Büroautomation zu berücksichtigen, da einerseits

- Linienbüros in Ausnahmefällen versagen, da sie sich nicht an Routineabläufen orientieren können, und andererseits
- Stabsbüros einer Ablaufunterstützung bedürfen.

1.5 INWIEWEIT IST BÜROAUTOMATION MÖGLICH UND WÜNSCHENSWERT?

Im folgenden soll versucht werden, die Auswirkungen von Aktivitäten im Bereich Büroautomation abzuschätzen und technisch sowie gesellschaftlich zu bewerten.

Der Fortschritt und die Verbreitung der Informationstechnik hängen in entscheidendem Maße von der Interessenlage der tragenden Institutionen ab; dies ist vornehmlich auf dem Gebiet der Rüstungstechnik zu beobachten, da hier wie in keinem anderen Bereich der Technik Entwicklungskosten kaum eine Rolle spielen. Büroautomation wird von Anwendern im industriellen und gewerblichen Bereich finanziert, die sich in folgenden Punkten

Fortschritte versprechen (vgl. insbesondere auch MERTENS85b (*)):

- (i) bessere Ausnutzung menschlicher Arbeitskraft,
- (ii) erhöhte Produktion pro Arbeitskraft,
- (iii) erhöhte Qualität der Arbeit,
- (iv) verringerte Durchlaufzeit einzelner Aufgaben durch die Organisation,
- (v) verbesserte Kommunikation zwischen den Arbeitenden einerseits und ihrem Führungspersonal andererseits (vgl. ABRAHAM81).

In welcher Hinsicht besteht nun die Möglichkeit einer technischen Unterstützung? Man kann hierbei vier Kategorien unterscheiden:

- *Arbeitsausführung.* Wie bereits erwähnt, sind hier sowohl die Quantität als auch die Qualität der erbrachten Arbeitsleistung steigerungsfähig. Durch die (Teil-)Automatisierung von Arbeitsfragmenten wird die Fehlerquote bei deren Ausführung verringert und durch eine größtmögliche Abkopplung von Bedienereingriffen die Unabhängigkeit von der menschlichen Arbeitsumgebung erreicht; so können aufwendige Arbeiten durchaus asynchron, z.B. in den Nachtstunden, erledigt werden. Es ist auch evident, daß an und für sich hochqualifizierte Arbeiten durch zunehmend minder qualifiziertes Personal erledigt werden können und Einarbeitungszeiten

(*) Die dort belegten Untersuchungen umfassen insbesondere auch eine Quantifizierung von Nutzeffekten.

kürzer werden - eine Erkenntnis, die in den beginnenden neunziger Jahren den hauptsächlichsten Diskussionsstoff der Sozialpartner abgeben dürfte.

- *Kommunikation.* Durch die Vielzahl von synchronen und asynchronen Kommunikationswegen innerhalb der Organisation und nach außen wird die Zahl an Dienstreisen und, mehr noch, die Zahl der Dienstbesprechungen drastisch verringert, die oft ganze Stäbe von kompetenten Mitarbeitern lahmlegen.
- *Informationstransparenz.* Information innerhalb des Betriebs verliert ihren Bezug zu der Stelle, an der sie endgelagert wird: Es sind, entsprechende Berechtigungen vorausgesetzt, Zugriffe auf alle Informationen innerhalb des organisatorischen Gesamtkomplexes möglich, ohne daß die heutzutage sehr schwerfälligen Mechanismen der Anforderung, der Kopie und des physischen Transportes an die anfordernde Stelle vonnöten sind. Gleichermaßen stehen globale Hilfsmittel zur Interpretation von Information zur Verfügung (Metainformation).
- *Infrastruktur.* Die Optimierung der Infrastruktur vollzieht sich unter der Prämisse, daß die Arbeitsumgebung des Bürotätigen voll auf ein Büroinformationssystem übertragen worden ist. Die resultierenden Vorteile sind:
 - *Verfügbarkeit* aller Hilfsmittel, seien es Taschenrechner, Telefonliste, passende Formulare oder anderer, leicht aus dem Blickfeld verlorener Gegenstände,
 - *fehlerlose Übertragung* von Information aus einer Repräsentationsform in eine andere, z.B. von der Telefonliste auf den Vermittlungsrechner, und

- Übertragung der Arbeitsumgebung an beliebige Stellen innerhalb des Betriebs.

Im Rahmen dieser Abhandlung darf in keinem Fall eine Liste der nachteiligen Auswirkungen eines Einsatzes von Büroautomationstechnik fehlen; man muß den meisten Veröffentlichungen aus diesem Bereich vorwerfen, daß sie häufig in blauäugiger Euphorie versinken und oft auch schlichtweg das Umfeld des Einsatzes dieser Techniken ignorieren (*).

- Arbeitslosigkeit. Die Behandlung dieses potentiell negativen Aspektes der Büroautomation ist Teil der immerwährenden Diskussion zum Thema *Rationalisierung* und wird häufig zum Fokus düster-pessimistischer Studien (siehe u.a. HINES79). Im Grunde ist der Kausalzusammenhang zwischen Mechanisierung/Automatisierung und Arbeitslosigkeit nichts Neues, und es ist eher eine Frage der Ethik des Arbeitsmarktes, ob ein Strukturwandel wie der betrachtete als eines der Fahrnisse des Arbeitslebens oder als von finsternen Verschwörern angezettelttes Attentat auf die arbeitende Bevölkerung

(*) Die im folgenden genannten Problemgebiete entsprechen in etwa dem Bereich, der in MERTENS85b (S. 78ff) durch die dort angeführte Liste von Thesen zu nicht-quantifizierbaren Befunden beim Einsatz von Informationstechnologie aufgespannt wird. Im einzelnen sind diese im Hinblick auf negative Auswirkungen:

- die Verarmungsthese,
- die Dequalifikationsthese,
- die Polarisierungsthese und
- die These vom Scheitern älterer Arbeitnehmer bei der Umstellung.

anzusehen ist - eine detaillierte Verfolgung dieses Problems dürfte jedoch nicht im Interesse des technisch orientierten Lesers sein.

- *Kontaktverlust.* Alle Bestrebungen gehen auf eine ausschließliche innerbetriebliche Kommunikation über elektronische Medien hin; Einsparungen an Dienstreisen und Besprechungen bringen den Verlust der Informationsübertragung und Einflußnahme durch unmittelbare Kontakte mit sich. U.a. sprechen Attewell und Rule in ihrer sehr optimistisch gefärbten Übersicht (ATTEWELL84) diesen Komplex an, ohne dabei zu einer eindeutigen Schlußfolgerung zu kommen.
- *Überbürokratisierung.* Unter übermäßiger Automatisierung wird in jedem Falle das Bild der Gesamtorganisation nach außen leiden. Die meisten Anwendungsumgebungen erlauben keine Reaktion des Systems nach ehernem Schema, vielmehr ist Flexibilität in der Einschätzung eines Sachverhalts sowie der Reaktion darauf gefragt (zu Anstrengungen hierzu vgl. Abschnitt 4 oder z.B. FIKES80).
- *Entqualifizierung.* Dieser Punkt wurde bereits oben erwähnt, muß hier jedoch nochmals, diesmal im Lichte der Arbeitnehmerseite, angeführt werden. Diese Gefahr wird auch in BULL86 als eines der "schwersten sozialen Probleme" angeführt.
- *Verlust der Privatsphäre.* Ähnlich wie im Fertigungsbereich bringt die Automatisierung und vollständige Erfassung von Vorgängen im Bürobereich auch die Möglichkeit einer erweiterten Überwachung der Arbeitsleistung mit sich. Während sich der Einsatz derartiger Systeme aufgrund gewerkschaftlicher Vorbehalte in Europa derzeit kaum durchsetzen läßt, haben

Feldversuche in den USA mit sog. *Computerized Performance Monitoring & Control Systems (CPMCS)* bereits erste Erkenntnisse über die Möglichkeiten und Auswirkungen dieses Einsatzgebiets erbracht (vgl. u.a. IRVING86). Grundsätzlich muß zunächst festgestellt werden, daß eine gerechte Bemaßung der Arbeitsleistung auf diesem Wege nur bei stereotypen bzw. reproduktiven Arbeiten des Linienbüros möglich ist, da im Stabsbüro wohl eher die Güte denn die Menge von Aktionen beurteilungswürdig ist. Die Einführung eines CPMCS bringt daher in jedem Falle eine Steigerung der Quantität der geleisteten Arbeit mit sich, eine Steigerung der Qualität kann nur im Fall systemseitig überprüfbarer Qualitätsnormen beobachtet werden, anderenfalls wirkt besagte Einführung eher *qualitätsmindernd* zugunsten der Quantität. Weiterhin sind eine höhere Streßbelastung, verringerte Zufriedenheit mit dem Arbeitsplatz und allgemein ein schlechteres Betriebsklima (Konkurrenz) zu beobachten. Als Positiva stellten sich die als objektiv empfundene Erfassung der Leistung der Mitarbeiter heraus, was im Zusammenhang mit der Erstellung individueller Leistungsbeurteilungen direkte Auswirkungen auf den Einzelmitarbeiter hat, gleichermaßen jedoch auch die sofortige und präzise Information des Managements über den Bestand an anstehenden und durchgeführten Arbeiten, analog zu Auftragsüberwachungssystemen im Fertigungsbereich.

Es ist jedoch keineswegs erforderlich, ein derartiges System als eigenständige Instanz zu installieren. Unter dem Deckmantel der hehren Absicht, die Kommunikation zwischen Arbeitsplatzrechner und Mensch zu verbessern, werden Abstraktionsprozesse mit dem *impliziten* Ergebnis einer Leistungs- und Kompetenzbewertung zur Grundlage der Adaption benutzergerechten Dialogverhaltens durchgeführt, die jedoch auch die Grundlage für Per-

sonalentscheidungen liefern können. Die technischen und organisatorischen Konsequenzen für den Schutz der Privatsphäre zeigt ROSENOW86 auf (S. 104ff).

- *Geistige Verarmung.* Mit der Bereitstellung umfassender Hilfsmittel und vorgeprägter Arbeitsabläufe geht eine geistige Verarmung des Bedienpersonals einher. Ein blindes Vertrauen in die Reaktionen der Maschine hemmt die Überwachungsfunktion des Bedieners und schaltet ihn als Regulans für ggf. falsche oder unangemessene Maßnahmen des Systems aus. Als warnendes Beispiel seien dabei nur die letzten 20 Jahre angeführt, in denen die Spezies *homo sapiens* infolge der weiten Verbreitung von Taschenrechnern das Kopfrechnen verlernte; dabei ist jedoch nicht die Existenz, sondern eher der unkontrollierte Gebrauch dieser Geräte anzuklagen. Man ist bereits heute oft Empfänger eines Schreibens, das nicht individuell erstellt, sondern aus Textkonserven zusammengefügt wurde; die Autoren umschreiben dabei den darzustellenden Sachverhalt gedanklich oft nicht mehr mit der eigentlichen Formulierung, sondern mit dem in ihrem Textverarbeitungswerkzeug gültigen Kürzel. Letztendlich führt dies zu einer stringenten *Sprachnormierung* mit all ihren Konsequenzen, die als Vision in ORWELL73 bereits ausgeführt wurden. Man könnte hier auch von einer *Kulturnormierung* im Sinne Douglas Hofstadters sprechen (vgl. HOFSTADTER80, S. 376).

"The Navy is a master plan designed by geniuses for execution by idiots (WOUKY2, S. 113)", ist die Feststellung eines der Protagonisten in Herman Wouks satirischem Roman zur amerikanischen Marine des 2. Weltkriegs. Indes, die hierdurch umrissene Situation, die für alle sozialen Einheiten, sei es Familie, Gemeinde oder Betrieb, charakteristisch ist, wird durch die

Automatisierung weiter verschärft werden. In vorgeschichtlicher Zeit konnte ein charismatisches und geistig reges Individuum seinen Einfluß auf bestenfalls 20 - 30 weitere Individuen ausüben (und dies nur mittelbar, d.h. durch Überzeugung oder Überredung), in der heutigen Zeit ist es möglich, daß der Autor eines Zeitungsartikels oder einer Fernsehsendung Millionen erreicht, oder gravierender, daß eine bestimmte Handlungsweise Millionen Bürgern auferlegt wird (z.B. Einkommensteuererklärung). Im Klartext: Moderne Kommunikationswege und Organisationswerkzeuge steigern den Verbreitungs- oder Beeinflussungsfaktor der Gedanken, Ideen oder Richtlinien einer privilegierten Gruppe auf die schweigende Mehrheit - "leaving people in only two roles: bosses and garbage collectors" (DRISCOLL82).

2 Die Datenstrukturen

"It is nothing but symbol chauvinism that makes computer scientists think that our structures are so much more important than material structures that (a) they should be perfect and (b) the energy necessary to make them perfect should be expended."

(DEMILLO79, S. 279)

Ziel dieses Kapitels ist die Erarbeitung des Begriffs *Dokument* sowie der auf ihm definierten Operatoren und der Gestaltung der Manipulationsschnittstelle. Dieser Begriff wurde im vorhergehenden Kapitel, namentlich bei der Bürodefinition in Abschnitt 1.1.1, intuitiv eingeführt und verwendet und soll in Intension und Gebrauch im folgenden verdichtet werden. *Dokument* wird dabei nicht im umgangssprachlichen Sinn - der gleichbedeutend ist mit dem juristischen Begriffsinhalt und eine Wertung der Form (*offiziell*) sowie der Bedeutung (*Glaubwürdigkeit, Tragweite*) impliziert - verstanden, sondern im Sinne von *Bürodokument*, d.h., als Ordnungsprinzip, verkörpert in einer allgemein anerkannten Architektur sowie Konventionen zu deren Benutzung in einer Büroumgebung. Die Vorschläge zur technischen Realisierung

orientieren sich an den in der Praxis auftretenden Forderungen, dabei u.a.

- Integration der verschiedenen Datentypen eines Dokuments,
- Integration der eingeführten Manipulationswerkzeuge in ein Dokumentenverwaltungssystem,
- Übernahme von Routinetätigkeiten und Reaktion auf Ausnahmesituationen (siehe dazu insbesondere Abschnitt 4).

Der Begriff Dokument konstituiert sich aus einer Komposition von untergeordneten Begriffen, nämlich den Gattungsbegriffen für die verschiedenen im Dokument vertretenen Repräsentationsformen von Information, die wiederum untereinander durch verschiedene Beziehungstypen verbunden sind. Die Beziehungstypen beeinflussen das Verhalten des Systems an der Mensch-Computer-Schnittstelle und das Erscheinungsbild des Dokuments insgesamt.

Dokumente sind als logische Konstrukte unabhängig vom Trägermedium zu verstehen. Als Beleg dient vorab die ISO-Definition dieses Begriffs:

"A document is defined as a material reproduction of the author's thoughts. The document's prime objective is to transmit, communicate and store these thoughts as accurately as possible, regardless of the medium used for these concepts."
(ISO84a, S. 5)

In diesem Kapitel wird nun schrittweise ein Dokumentenmodell entwickelt, das versucht, insbesondere die dynamischen, und damit visuellen, Aspekte der Dokumentenbearbeitung zu berücksichtigen und dabei möglichst geringe Anforderungen an die bearbeitenden Programme zu stellen. Die Betrachtungen basieren auf der Hypothese, daß der ausschließliche Bezug von Software

aus einer Quelle im Bürobereich illusorisch ist und dennoch das Kompatibilitätsniveau der derzeit im Einsatz befindlichen Büroprogramme erschreckend niedrig ist. Der Vorschlag des Dokumentenmodells bietet den Übergang zur Skizzierung eines auf analogen Prinzipien basierenden Systems zur Verwaltung der Benutzerschnittstelle. Dabei werden neben der Verdeutlichung der im Büro anzutreffenden Primär-Datenstrukturen auch die unter der obengenannten Prämisse erforderlichen Meta-Datenstrukturen angegeben.

2.1 DIE BASISDATENTYPEN

Typenbildung im konstruktiven Sinne bezeichnet die Ausführung eines Abstraktionsoperators auf eine Ansammlung von Objekten (in der Literatur oft auch mit dem Terminus *entity* bezeichnet). Im Sinne von CARDELLI⁸⁵ (S. 473) gehen dabei sowohl Aspekte der Verwendung eines Datenobjekts, d.h. seiner Auswirkungen auf andere Objekte des Objektuniversums, als auch Aspekte des Verhaltens des Objektes, namentlich der Änderung seines internen Zustandes als Kriterien in den Abstraktionsoperator ein. Die entstehende Untermenge bezeichnet dabei die *Extension* des Typs, die sich ergebende interne Struktur zusammen mit der Beschreibung des Verhaltens aller Elemente der Untermenge die *Intension*. Die Motivation dabei ist

- durch Abstraktion bereits möglichst frühzeitig typinhärente Eigenschaften überprüfen zu können,
- effiziente Verfahren zur Bearbeitung von typgebundenen Objekten ermitteln zu können,

- von einer Ablage von Metadaten pro Objekt auf eine Ablage von Metadaten pro Typ übergangen zu können.

Welche Aspekte eines Datentyps sind in der Datenverarbeitung interessant? Ähnlich wie im Bereich der Datenbanksysteme ist auch hier eine Betrachtung auf drei Ebenen möglich (vgl. Drei-Schema-Architektur nach ANSI/SPARC in WEDEKIND81 S. 72f; siehe auch Abb. 2-1):

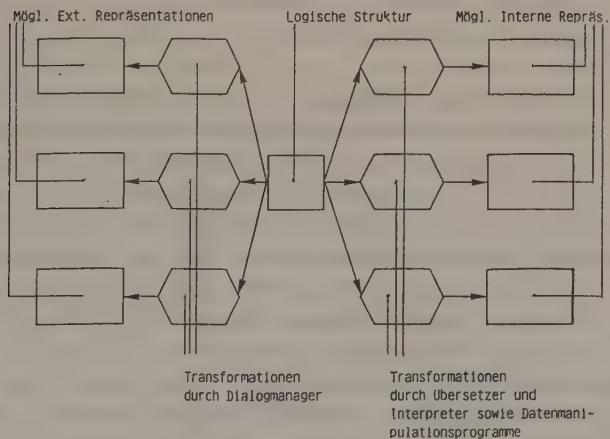


Abb. 2-1: Ebenenmodell des Datentyps

- *Logische Struktur:* Sie beschreibt den dynamischen Aspekt eines Datentyps, d.h. die auf einem Typ definierten Operatoren. Dabei sind die Zugriffsoperatoren (Selektoren) zum Zugriff auf Teilobjekte eingeschlossen.
- *Interne Repräsentation:* Hierbei werden die Repräsentation der Objekte eines Typs auf Speichermedien und die unteren Ebenen der entsprechenden Zugriffprozeduren definiert. Es wird dabei in jedem Falle eine Unterscheidung hinsichtlich der Charakteristika der in Frage kommenden Speichertypen geben.
- *Externe Repräsentation:* In der herkömmlichen Programmierung wurde dieser Aspekt oft dem Anwendungssystem überlassen, das diese Repräsentation ad hoc vornahm. In zeitgemäßen Programmierungsumgebungen ist inzwischen eine isolierte Darstellbarkeit und Manipulierbarkeit von Datenobjekten erforderlich, so daß Darstellungsform, Parametrierungsform, Zugriffsverhalten u.ä. explizit formuliert werden müssen.

In den folgenden Teilabschnitten werden die Basisdatentypen von Dokumenten unter diesen Aspekten behandelt. Die Beschreibung erfolgt informell; es kann nicht Aufgabe dieser Abhandlung sein, isolierte Datenstrukturen zu spezifizieren, eher, aus Phänomenen der Anwendungswelt zu einer ganzheitlichen Betrachtungsweise zu kommen. Die Beschränkung auf die *grundlegenden* (Basis-) Typen in den folgenden Unterabschnitten bedeutet eine vorläufige Ausgrenzung von dokumententypischen Strukturen, z.B. Fußnoten, Überschriften etc.

2.1.1 ~~TEXT~~

Die Erschließung von *Text* für datenverarbeitende Geräte war der erste Schritt auf dem Wege zur Büroautomation. Das grundsätzlich Neue an der Textverarbeitung waren die Abkehr von der satzorientierten, *formatbehafteten* Datenstrukturierung und die Hinwendung zur *formatfreien* und deshalb bis zu gewissem Grad interpretativ zu verarbeitenden Datenstruktur.

Logische Struktur. Text wird als Aneinanderreihung von Zeichen aus einem vordefinierten Alphabet verstanden. Entgegen der standardisierten Terminologie, die alle für die menschliche Wahrnehmung geeigneten Darstellungsformen einbezieht (ISO84a), werden nur alphanumerische Informationstypen im Sinne einer schriftlichen Niederlegung betrachtet. Entsprechend dem Sprachgebrauch sind dabei

- *Worte* als Zeichenketten zwischen zwei Begrenzungszeichen (*Delimiter, white space*),
- *Sätze* als Zeichenketten zwischen Satzabschlußzeichen (*Punkt, Strichpunkt, Doppelpunkt, Ausrufezeichen, Fragezeichen*) und
- *Abschnitte* als Zeichenketten zwischen Abschnittabschlußzeichen (meist spezielles Symbol)

bekannt. Die üblichen Basisoperationen (Einfügen, Ändern, Löschen) sind für alle derartigen Untertypen eingeführt. Hinsichtlich der Selektion müssen die Prädikate

enthält (text#, zeichenkette)

bzw.

ist-relevant-für (text#, begriff)

definiert sein. Darüber hinaus kann *enthält* noch durch Subprädikate ergänzt sein, die positionale Eigenschaften innerhalb des Textes abdecken.

Externe Repräsentation. Das äußere Erscheinungsbild von Text wird durch die sog. Layout-Parameter bestimmt. U.a. finden dabei folgende Eigenschaften Eingang:

- Formatierungsart (Rechts-, linksbündig, Blocksatz, Flattersatz, mittig),
- Default-Alphabet,
- Äußere Abmessungen.

Weiterhin sind Textobjekte *paginierbar*, d.h., in manchen Fällen kann eine externe Aufspaltung eines Textobjektes in zwei Blöcke vorgenommen werden. Ähnliches gilt hinsichtlich der *Trennung* in bezug auf Worte, wobei hier als zusätzliche Komplikation noch die Abhängigkeit der Menge der möglichen Trennungen vom *Textinhalt* auftritt. Als weitere Manipulationseinheit der externen Repräsentation ist die *Zeile* anzusehen, die sich nach Anwendung der Layout-Parameter auf den Text konstituiert.

Interne Repräsentation. Die Ablage von textuellen Objekten erfolgt im allgemeinen unformatiert in Dateien bzw. im Datenbankbereich in *Langen Feldern* (*long fields*). In den meisten Fällen sind Layout-Parameter nicht explizit angegeben, sondern eher implizit in Form von Zeichensequenzen, deren Semantik erst zum Ein-/Ausgabezeitpunkt interpretativ ermittelt wird (z.B. WordStar-Format). Als Beispiele für die explizite Verwaltung von Text- und Layouteigenschaften seien die Smalltalk-Umgebung (vgl. u.a. TESLER81, GOLDBERG83), das interne Format des STAR-Systems (HAFFER85), ein Editor auf Basis des Lilith-Arbeitsplatzrechners

(GUTKNECHT85) und ein Ansatz für datenbankorientierte Erschließung von Texten (vgl. STONEBRAKER83c) genannt.

In der praktischen Anwendung trifft man Hilfsstrukturen zur Unterstützung der Dokumentselektion (Information Retrieval) an. Die dabei angewendeten Methoden sind entweder schlüsselwortgebunden (Deskriptoren, Thesauren) oder für die Wiedergewinnung von Freitext (Invertierung von Fragmenten, Signaturen) geeignet. Eine gute Übersicht der dabei angewendeten Techniken gibt FALOUTSOS85.

2.1.2 GRAFIK

Der Begriff Grafik steht umgangssprachlich für eine im Gegensatz zu naturalistischen Darstellungsformen bis zu gewissem Grade abstrahierte Darstellung realer Objekte, beziehungsweise die Darstellung abstrakter Objekte an sich. Im Bereich der Datenverarbeitung wird der Begriff zu Computergrafik spezialisiert. In ihrer paradigmatischen Arbeit (FOLEY82) definieren Foley und Van Dam Computergrafik als "... the creation, storage, and manipulation of models of objects and their pictures via computer" (FOLEY82, S. 3). Die weiteren verfügbaren Definitionen befassen sich in erster Linie mit den technischen Aspekten der Repräsentation bildlicher Daten und führen als Motivation meist die alte Lebensweisheit "Ein Bild sagt mehr als tausend Worte!" an.

Logische Struktur. Computergrafik basiert auf einem Grundmodell der bildlichen Darstellung, das wiederum die Vereinigung zweier disjunkter Darstellungsmodelle beinhaltet. Man spricht dabei vom

- *Rastermodell*, bei dem ein Bild aus einem Raster von Bildpunkten zusammengesetzt ist (u.a. zur Wiedergabe von Fotografien), und dem
- *Kalligrafischen Modell*, das eine Grafik aus Vektoren, Kegelschnitten und Punkten mit Attributierung der Linienführung zusammensetzt; man spricht in diesem Zusammenhang auch oft von *Vektormodell*. Im allgemeinen sind Vektordarstellungen flexibler in puncto Maßstabsveränderung und Zerrung/Streckung (*).

Die Vereinigung und dynamische Veränderung von Objekten dieser logischen Repräsentationsformen sind Gegenstand umfassender Normierungsbemühungen. Hervorzuheben ist dabei das *Graphische Kernsystem* (GKS), dessen Grundzüge u.a. in ENDERLE83, GORNY84 und BONO85 dargestellt sind; diese inzwischen auf ISO-Niveau anerkannte Norm nimmt jedoch auch Aspekte der Präsentation und Konfiguration des Tragersystems vorweg, so daß in diesem Rahmen auf eine detaillierte Darstellung verzichtet werden soll.

Ein weiteres Gestaltungsmittel im Rahmen der logischen Struktur sind *Farbe* und *Flächenstruktur*. Dabei wird in erster Linie der Kontrast verschiedener Farben bzw. Struktur vermerkt, die Zuordnung eines Farbwertes erfolgt zur Laufzeit durch Auswahl aus Farb- bzw. Strukturtabellen.

Externe Repräsentation. Diese ist in erster Linie von der zugrundeliegenden Hardware abhängig. Man hat sich in letzter Zeit in steigendem Maße bemüht, bereits bei normierten Verfahren eine Abtrennung der Externen Repräsentation zu erreichen und so weitgehende Geräteunabhängigkeit erzielen können. Die wesentlichen Merkmale der Abbildung auf die Externe Repräsentation sind:

(*) Der Fall der Darstellung dreidimensionaler Objekte (*Volumenmodell*, *Drahtmodell*) wird hier übergangen.

- Abbildung von Darstellungselementen und Interaktionen gemäß Geräteeigenschaft, d.h. Berücksichtigung von Farbfähigkeit und ähnlichen Attributen und ggf. Ersatzdarstellungen dafür sowie Einbezug von speziellen Eingabegeräten (Zeigeeinstrument, vgl. hierzu auch NEWTON85, S. 29ff)
- Koordinatentransformation, d.h. Transformation aus einem quasi-unendlichen Weltkoordinatensystem in ein gerätespezifisches Koordinatensystem.

Interne Repräsentation. Ähnlich wie Text wird Grafik zumeist in werkzeugspezifischer Repräsentation abgelegt und ist somit von der Interpretation der bearbeitenden Werkzeuge abhängig. Werkzeugübergreifende Standards wie z.B. GKSM (GKS-Metafile) setzen sich jedoch in steigendem Maße durch. Der Wert einer derartigen Standardisierung wird selbstverständlich durch den Einsatz von gerätetypischen Primitiven, ausgedrückt durch die unschuldige Formulierung *Verallgemeinertes Darstellungselement*, bzw. des Primitivs *Zellmatrix* stark gemindert, da in diesen Fällen die Hardware des erzeugenden Geräts mit das Layout und die Aussage der Grafik bestimmt.

2.1.3 TABELLARISCHE INFORMATION

Tabellen sind wohl die älteste Organisationsform von Massendaten schlechthin. Mit der Einführung elektronischer Verarbeitungsverfahren wurden die in den Anwendungswelten vorhandenen Tabellenstrukturen direkt auf entsprechende Datenstrukturen (*array*, *file*) abgebildet.

Logische Struktur. Tabellen sind 1- bis n-dimensionale Datenstrukturen, wobei das Einzelelement (Zelle) durch die Vereinigung der aufgrund seiner Lage zutreffenden Achsenbeschriftungen interpretierbar wird. Im einfachen und im Alltag am häufigsten auftretenden Fall ist dies eine Werteleiste, der eine entsprechende Beschriftungsleiste gegenübersteht, bzw. eine Matrix mit zwei Beschriftungsleisten.

In der Praxis ist der einer Beschriftung zuzuordnende Wertebereich oft nach Achsen organisiert, d.h., eine Anzahl von Achsen innerhalb der Tabelle konstituiert die darin auftretenden möglichen Werte, während die anderen die entsprechenden Ausprägungen parametrisieren. Dies entspricht bei einer n-dimensionalen Tabelle der Auswertung einer Funktion

$$f(x_0, \dots, x_{n-1}) = x_n,$$

wobei der Wertebereich $d(f)$ die Vereinigung der Wertebereiche $d_0(f)$ bis $d_k(f)$ mit $0 < k < n$ darstellt.

Im allgemeinen Fall ist eine derartige Achsenorientierung nicht gegeben; der Wertebereich einer Zelle richtet sich nach der Gesamtheit aller Parameter. Der Wert einer Zelle wird folgendermaßen bestimmt:

- durch willkürliche Werteingabe (mit und ohne intuitiven Bezug zur Achsenbeschriftung),
- durch funktionale Abhängigkeit von Achsenbeschriftungen,
- durch funktionale Abhängigkeit von Achsenbeschriftungen sowie den Werten anderer Zellen.

Dabei hängt die Konsistenz einer Tabelle von folgenden Kriterien ab:

- *Wertebereich:* Der Inhalt einer Zelle muß zu jedem Zeitpunkt dem für sie definierten Wertebereich entsprechen.
- *Abhängigkeit:* Bei Abhängigkeit des Zelleninhaltes vom Inhalt anderer Zellen bzw. der Achsenbeschriftungen muß der Wert der Zelle gemäß dem mit ihr assoziierten Ausdruck diesen funktional übergeordneten Zellen entsprechen.

Externe Repräsentation. Für die externe Darstellung von Tabellen können folgende beiden Grundformen unterschieden werden:

- *Tabellarische Ausgabe:* Dabei entspricht die externe Darstellung vom Erscheinungsbild dem intuitiven Verständnis von Tabellen, wobei jedoch vernünftigerweise nur zweidimensionale Tabellen derart darstellbar sein sollten.
- *Grafische Ausgabe:* Tabellen werden mit Mitteln der sog. Bürografik (*Business Graphics*) veranschaulicht.

Letztere (vgl. hierzu u.a. MORBEY88, S. 23ff) verfolgt die analoge Darstellung von Größen und Größenverhältnissen mit Hilfe grafischer Darstellungselemente. Dabei können auch Farbe und Schraffur in die Darstellung miteinbezogen werden. Die in der Bürografik meistverbreiteten Basistypen sind

- o Liniendiagramme,
- o Balkendiagramme,

o Torten- bzw. Kreisdiagramme.

Weitere Typen, die zumeist auf diese Grundtypen übergeführt werden können, nennt NEWTON85 (S.43ff). In Einzelfällen kann dabei auch noch eine dritte Dimension durch perspektivische Darstellung wiedergegeben werden; im allgemeinen ist davon jedoch abzuraten.

Die grafische Darstellungsmöglichkeit für tabellarische Information ist zu einer der Schlüsseleigenschaften moderner Büroautomationstechniken geworden. Man muß jedoch die Vorteile dieser überbewerteten Fähigkeit relativieren: Grafische Darstellung tabellarischer Daten bringt nur Vorteile (vgl. dazu u.a. DICKSON86), wenn:

- (i) in erster Linie Muster, Verläufe zu erkennen bzw. Größenvergleiche anzustellen sind und
- (ii) sehr starke zeitliche Restriktionen (Vorträge, Massenauswertung) bestehen.

Hingegen ist die tabellarische Darstellungsweise überlegen, wenn

- (i) die Datenmenge überschaubar ist und
- (ii) die Extraktion exakter Werte von großer Bedeutung ist.

Interne Repräsentation. Tabellen können auf alle gängigen Datenmodelle im Datenverwaltungsbereich abgebildet werden. Besonders geeignet scheint dabei das relationale Datenmodell, das den meisten neueren Datenbanksystemen zugrundeliegt. Diese Feststellung muß jedoch hinsichtlich der oben erwähnten Abhängigkeiten von Zellen abgeschwächt werden: Die flexible Definition und Überwachung von Konsistenzbedingungen und Abbildungsvorschriften ist bisher noch in keinem marktüblichen System

realisiert und muß vom Anwendungsprogramm (Tabellenkalkulationsprogramm, Tabelleneditor) selbst geleistet werden. An der internen Ebene eines derartigen Systems können wiederum je nach Tabellenaufbau neben den herkömmlichen Invertierungstechniken (B-Bäume, Hash-Verfahren etc.) spezielle Techniken zum Einsatz kommen (z.B. für kontinuierliche Achsenabschnitte die Grid-File-Technik, vgl. HINRICHS83).

2.1.4 AUDIO-VISUELLE INFORMATIONSTYPEN

Die im folgenden erwähnten Informationstypen spielen in den weiteren Ausführungen keine bedeutende Rolle; für die Erschließung des Dokumentbegriffs im Büro sind sie auch nicht von Bedeutung, da sie auf bereits behandelte Typen rückführbar sind.

Akustische Information (Digitalisierte Sprache). Die Ablage, Bearbeitung und Wiedergabe akustischer Information, dabei in erster Linie gesprochener Worte, ist in folgenden Bereichen von Bedeutung:

- *Annotationen:* In einigen meist experimentellen Systemen ist an beliebigen Stellen das Einfügen von Bemerkungen möglich, wobei diese aber nicht vom System interpretiert, sondern nur unverändert an einen späteren Betrachter weitergegeben werden müssen. Das Vorhandensein einer Annotation wird durch ein Symbol an der entsprechenden Stelle gekennzeichnet; durch Bezugnahme auf dieses Symbol kann das Abspielen einer Bemerkung ausgelöst werden.

- *Integration von Telefondiensten in Kommunikationssysteme:* Die Erfassung digitalisierter Sprache bewirkt die Integration in herkömmliche inner- und zwischenbetriebliche Kommunikationssysteme. Durch die Eindeutigkeit von Telefonnummern ist ein Abgleich mit herkömmlichen Adressierungsschemata in rudimentärer Form möglich. Ein Beispiel für ein derartiges System ist ADS (Audio Distribution System) von IBM (siehe hierzu RICHARDS82, über die Feldversuche berichtet GOULD83).

In beiden Fällen wird die Sprache nach der Digitalisierung *uninterpretiert* abgelegt. Die einzigen definierten Operationen sind Ablage und Löschen des Sprachobjektes sowie die Übergabe an die Treiberschnittstelle eines Sprachsynthesegerätes zwecks Wiedergabe. Das Edieren derartiger Objekte kann nur durch Definition einer Struktur auf diesen Objekten erfolgen; dabei wären einerseits die *explizite* Strukturierung (z.B. durch Drücken einer Funktionstaste) in benutzerbestimmte Einheiten (z.B. Sätze oder Abschnitte) oder die *automatische* Strukturierung in Worte bzw. Morpheme denkbar. Für marktfähige Systeme mit dieser Eigenschaft ist allerdings ein gesichertes *Sprachverstehen* erforderlich, so daß in absehbarer Zeit wohl nicht damit gerechnet werden kann. Derzeit verfügbare Sprachbearbeitungssysteme verwenden zur Traversierung ein Grobraster vom Benutzer definierter logischer Einheiten und ein Zeitraster als Feinsteuerung (siehe z.B. NICHOLSON85).

Animation. Dieser Darstellungsform liegt der Gedanke zugrunde, durch die Definition von Bildfolgen statt statischer Bilder komplexe Sachverhalte auszudrücken. Insbesondere ist Animation dazu geeignet, die Darstellung von tabellarischen Datenstrukturen durch Bürografik, die im praktischen Einsatz auf drei Dimensionen beschränkt war, um eine weitere Dimension zu ergänzen. Die auf Bildfolgen definierten Funktionen entsprechen denen der digitalisierten Sprache, wobei in diesem Falle auch noch Geschwindigkeitsaspekte eine Rolle spielen können. Die interne

Repräsentation ist bemüht, eine Ablage in Form von voll repräsentierten Bildern mit Folgenummern zu vermeiden und eher ein abstraktes Modell für die sukzessiven Veränderungen von Bildern zu finden (wie dies bei einer vierdimensionalen Tabelle mit entsprechender Attributierung bereits möglich ist). Abschließend kann nur auf die derzeit und in nächster Zukunft noch geringe Bedeutung dieser Technik verwiesen werden: für anwendungsneutrale Arbeitsplatzrechner sind Prozessor- und Speicherbedarf für Bildfolgen nach dem heutigen Stand der Technik zu groß; Animation im eingeschränkten Sinn (Pan-, Zoom- und Rotationsoperationen, auch für dreidimensionale Grafik) wird in erster Linie im CAD-Bereich angewendet.

2.2 BEARBEITUNGSWERKZEUGE (BÜROBASISPROGRAMME)

Die im vorhergehenden Abschnitt angeführten Basistypen sind die Summe der geschichtlichen Entwicklung der Büroautomation. Im Zuge dieser Geschichte wurden einige Programmsysteme, insbesondere im Mikrocomputerbereich, zu den Archetypen für Bearbeitungswerkzeuge ihrer Kategorie schlechthin, die in der Literatur auch als *Bürobasisprogramme* bezeichnet werden. Man denke dabei u.a. an *WordStar* für den Textbereich oder *VisiCalc* und *Multiplan* für den Bereich der Tabellenverarbeitung. Ziel dieses Abschnittes ist die Erläuterung der Grundprinzipien dieser Werkzeuge, ohne auf die Details der Einzelimplementierung einzugehen. Der dabei eingeführte Begriff *Werkzeug* steht für die Summe aller Funktionen, die für den bearbeiteten Datentyp definiert sind, sowie des Dialogverhaltens. Letztendlich können alle Werkzeuge als *Edierwerkzeuge* betrachtet werden, die jedoch im Gegensatz zu den herkömmlichen Werkzeugen dieser Art (auf der Ebene der Dateien und Zeichenketten) komplexere Datentypen und entsprechende Funktionen anbieten. Als Funktionsgruppen lassen sich dabei folgende

Kategorien unterscheiden:

- *Edieren*: Suchen, Eintragen, Verändern und Löschen von Information.
- *Speicherverwaltung*: Abspeichen und Sichern von Information auf Sekundärspeichermedien.
- *Ausgabefunktionen*: Ausgabe auf Druckgeräten.
- *Verwaltungsfunktionen*: Verwaltung von Zugriffsberechtigungen, Übertragung von Information in kompatibeltem Format an andere Werkzeuge.

Verfolgt man die Analogie zum Edieren weiter, werden folgende Komponenten eines Edierwerkzeugs, und damit eines Basisprogramms im allgemeinen, sichtbar (vgl. hierzu auch MEYROWITZ82a, S. 328ff):

- die *Betrachtungskomponente* (*viewing component*), die, gesteuert durch einen Filtermechanismus, einen Ausschnitt aus einer gemäß dem Bearbeitungsmodell organisierten Datenstruktur für die Betrachtung aufbereitet,
- die *Edierkomponente* (*editing component*), die durch einen eigenen Filtermechanismus Teile der Datenstruktur bearbeitet (verändert, löscht),
- die *Traversierungskomponente* (*traveling component*), die die Betrachtungs- und Edierausschnitte verändert und Suchvorgänge, entweder nach Suchkriterien oder durch Blättern (*browsing*), ermöglicht,

- die Darstellungskomponente (display component), die die Abbildung auf ein physisches Gerät sowie ggf. den Abgleich mehrerer Darstellungsflächen durchführt.

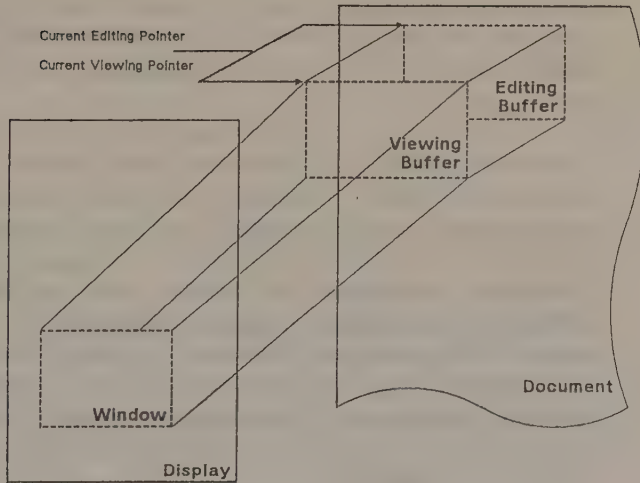


Abb. 2-2: Idealisierte Abbildung in einem Edierwerkzeug
(aus MEYROWITZ82a, S. 331)

Abb. 2-2 zeigt idealisiert die Abbildung eines Ausschnittes aus einer Datenstruktur, hier allgemein und unpräzise als *Dokument* bezeichnet, durch die Edier- und Betrachtungskomponente bis zur Einblendung auf einem Ausgabegerät durch die Darstellungskomponente. Die behandelten und in der Darstellungskomponente repräsentierten Datenstrukturen sind als abstrakte Datentypen bzw. Objekte im Sinne von GUTTAG77 aufzufassen, die durch die

Funktionen des Werkzeugs manipuliert werden können.

Hinsichtlich der Integrationsfähigkeit von Bürobasisprogrammen lassen sich in folgenden Bereichen gravierende Mängel feststellen:

- *Benutzerschnittstelle:* Bis auf wenige Ausnahmen sind die Benutzerschnittstellen nicht anpassungsfähig und deshalb auch nicht voll in eine Gesamtlösung integrierbar.
- *Abgeschlossenheit:* In den meisten Fällen müssen die Funktionen anderer Bereiche durch ein Werkzeug mitübernommen werden, was oft zu unbefriedigenden Lösungen führt. Ein Beispiel dafür sind die interaktiven Textsysteme, die häufig über primitive grafische Möglichkeiten verfügen. In vielen Fällen hat gerade der Erfolg eines Werkzeugs in seinem Bereich dazu geführt, daß, in unseriöser Weise, ein Spezialwerkzeug zu einem Universalwerkzeug mutiert wurde. Die interaktive Integration anderer Werkzeuge wird dabei kaum unterstützt.
- *Formate und Datenaustausch:* Infolge fehlender, verfehlter oder zu spät eingeleiteter Normierung auf dem Gebiet der Büroautomation (abgesehen vom Kommunikationsbereich) definieren diese Werkzeuge ihr eigenes logisches und internes Datenmodell. Daher ist die Konvertierung von Datenobjekten nur eingeschränkt möglich, und diese muß noch dazu im Einzelfall programmiert werden.

Im Falle einer datenstrukturorientierten Betrachtungsweise entsprechen die werkzeugspezifischen Repräsentationsformen einer nach einem werkzeugspezifischen Datenmodell organisierten Datenmenge. Natürlich drängt sich der Gedanke auf, dafür ein Datenmo-

dell höherer Ordnung einzusetzen. Aufgrund der kraß divergierenden Beurteilungen des Nutzens dieser Modelle sowie ihrer relativ geringen Verbreitung ist jedoch als gemeinsamer Nenner die sequentielle Datei das maximal Erreichbare, wobei diese zudem auch durch praktisch alle gängigen Betriebssysteme unterstützt wird.

2.2.1 TEXTVERARBEITUNG

Das Problem der Textverarbeitung bzw. -formatierung berührt zwei Teilaufgaben:

- Definition des Inhalts und der Struktur von Text,
- Erzeugung einer Wiedergabe des so definierten Inhalts gemäß bestimmter Regeln und Direktiven.

Bei Textverarbeitungswerkzeugen lassen sich grob zwei Klassen unterscheiden:

- o WYSIWYG-Werkzeuge nach dem Entwurfsprinzip "What You See Is What You Get" erlauben die direkte Manipulation des ausgabefertigen Dokuments. Es ist der Grundgedanke dieses Prinzips, daß die Erscheinungsbilder auf der Bildschirmoberfläche und nach Bearbeitung auf dem Papier identisch sind. Für die Implementierung der Werkzeuge erfordert dies, daß die Formatierungsalgorithmen folgende Anforderungen erfüllen:

- *Interaktives Formatieren:* Nach jeder Interaktion muß der bearbeitete Text dem endgültigen Erscheinungsbild entsprechen.
- *Inkrementelles Formatieren:* Der Formatierer muß für jede Interaktion den Bereich, der damit invalidiert worden ist, erkennen und nur für diese Bereiche eine Neuformatierung anstoßen. Beispielsweise wird durch das Einfügen eines Zeichens in jedem Fall eine Neuformatierung der aktuellen Zeile nötig; bei Zeilenüberlauf, und damit -neuumbbruch, müssen der aktuelle Abschnitt und eventuell sogar die Seite neu umgebrochen werden.
- *Verzögertes Formatieren:* Interaktionen sollen zu jedem Zeitpunkt möglich sein; Umbruchoperationen werden ggf. zugunsten der Ein-/Ausgabeverwaltung zurückgestellt.

Die Meta-Informationen, wie z.B. Kopf- und Fußzeilenbegrenzung oder Randbereiche, sind dabei dem Erscheinungsbild auf dem Bildschirm überlagert. Es ist für derartige Systeme charakteristisch, daß, sofern es sich um Systeme mit mehreren Fonttypen und Proportionalsschrift handelt, eine extrem hohe Belastung der Zentraleinheit sowie der Ein-/Ausgabekanäle zu beobachten ist, aber auch, daß die Strukturierungs- und Typisierungsmöglichkeit für Textbausteine äußerst eingeschränkt ist. Derartige Systeme sind jedoch sehr wohl geeignet für wenig strukturierte Texte (Briefe, kurze Berichte), aber jedoch kaum für umfangreiche publizistische Arbeiten. Eine sehr gute Übersicht über den Stand der Technik des Edierens unter Berücksichtigung von WYSIWYG-Textsystemen findet sich in MEYROWITZ82a/b.

- o Mark-up-Werkzeuge dienen zur Bearbeitung von Dokumenten im Batch-Betrieb. Der zu formatierende Text wird mit eingestreuten Formatierungsdirektiven markiert; dadurch kann nicht nur das äußere Erscheinungsbild, sondern auch die Intention eines Textabschnittes abgelegt werden. Für die praktische Anwendung heißt dies: Texte können eher in logische Einheiten aufgeteilt werden, der Benutzer orientiert sich, im Gegensatz zu WYSIWYG, weniger an der äußeren Erscheinungsform des Textes, als am logischen Aufbau. Modellfall für den Einsatz von Mark-up-Formatierern ist die verteilte Erstellung eines Dokuments, wobei ein Satz an Standardbausteinen (Kapitel, Fußnote, Anmerkung) den einzelnen Autoren zur Verfügung gestellt wird sowie deren Gebrauch reglementiert wird. Details der Externen Repräsentation werden dabei erst zu einem späteren Zeitpunkt festgelegt, so daß diesbezüglich keinerlei Berücksichtigung durch die Autoren stattfinden kann, und deshalb ein einheitliches Erscheinungsbild des Gesamtdokuments wieder durch Beeinflussung an einer zentralen Stelle erfolgen kann. Dabei werden dann die logischen Attribute in physische Attribute auf der Ebene der Externen Repräsentation übergeführt. Ein weiteres Argument für den Einsatz von Markup-Formatierern ist die Unabhängigkeit von Restriktionen, die durch den Dialogbetrieb bedingt sind: Die relative Positionierung einzelner Buchstaben sowie der Umbruch im Verband übergeordneter logischer Einheiten erfordert relativ komplexe Algorithmen, die im WYSIWYG-Betrieb zu inakzeptablen Antwortzeiten führen wurden.

Standardisierungsbemühungen auf diesem Gebiet schlugen sich in der Definition einer *Document Markup Metalanguage* durch die International Standards Organization nieder (ISO84b). Die in der Textverarbeitung gebräuchliche Bezeichnung ist hierfür *Standard*

Generalized Markup Language (SGML).

Neben der expliziten Markierung werden auch Verfahren zur impliziten *inhaltsbezogenen* Markierung anhand von Strukturelementen (z.B. Interpunktion) eingesetzt, die im Text selbst enthalten sind (siehe u.a. PEEL85).

Eine Übersicht über die heute gebräuchlichen Batch-Formattersysteme bietet FURUTA82.

In einigen Fällen bemüht man sich, Experimentalsysteme mit hybrider Arbeitsweise zu erstellen und so die Vorteile beider Ansätze zu vereinen. Eine Darstellung der dabei auftretenden Probleme anhand eines konkreten Projekts findet sich in CHAMBERLIN82 und CHAMBERLIN85.

2.2.2 **TABELLENKALKULATION**

"A spreadsheet is a simulated pocket universe that continuously maintains its fabric; it is a kit for a surprising range of applications."

(KAY84, S. 57)

Tabellenkalkulationsprogramme bearbeiten, wie der Name schon sagt, den Basisdatentyp *Tabelle* (vgl. Abschnitt 2.1.3). Die Assoziation an betriebliche Kalkulationsverfahren verdanken sie der von der Verfahrensweise her äußeren Ähnlichkeit mit den in früheren Zeiten angewandten papier- und formulargestützten

Kalkulationsverfahren. Den Übergang auf die dynamische Verarbeitung mittels eines Rechners verdanken sie der Frustration von Daniel Bricklin und Robert Frankston bei der Anwendung herkömmlicher Verfahren; sie schufen das Werkzeug *VisiCalc*, das ihnen wirtschaftlichen Erfolg und ihren Kunden neben einem flexiblen Kalkulationswerkzeug auch weitere Einsatzmöglichkeiten als Prognose-, Simulations- und Abrechnungswerkzeug öffnete (vgl. dazu KAY84, S. 57).

Eine *VisiCalc*-Tabelle setzt sich aus einer Menge von Zellen zusammen, die kartesisch angeordnet sind und mit Hilfe eines darauf aufbauenden Koordinatensystems adressiert werden können. Die Abmessungen einer Tabelle sind konzeptionell unbeschränkt, jedoch meist durch technische Randbedingungen (Hauptspeichergröße) eingeschränkt. Dabei muß beachtet werden, daß derartige Programme meist auf Kleinstrechenanlagen (Personal Computer u.ä.) ablaufen, da derartige Programme mit ihren aufwendigen Benutzerschnittstellen nur schwer in das Dialogschrittdenken der Großrechnerwelt eingepaßt werden können.

Entscheidend für die Gesamtfunktionalität des Gesamtprogramms ist die Funktion der einzelnen Zelle: gemeinhin sind hierfür zwei Möglichkeiten vorgesehen:

- Durch die Vorgabe von *Konstanten* wird einerseits das Bild der Gesamttabelle geprägt (Beschriftungen) und andererseits werden die Berechnungen selbst initiiert (Vorgabewerte).
- Durch Vorgabe von *Verfahren* bzw. *Regeln* werden Abhängigkeiten von Zellen untereinander konstituiert. Eine derart gekennzeichnete Zelle verändert ihren Wert, wenn sich die Werte der Zellen verändern, die in die ihr zugrundeliegende Regel eingehen. Dieser ggf. kaskadenartige Effekt kann je nach der Implementie-

rungsansatz bei manchen Eingaben zu längeren Wartezeiten führen.

2.2.3 GRAFIKEDITOREN

Grafikeditoren (drafting packages) dienen der Erstellung von bildlichen Darstellungen für den Bürogebrauch und basieren deshalb meist auf sehr einfachen geometrischen Modellen. An dieser Stelle sollte eine Abgrenzung zu den im CAD-Bereich gebräuchlichen Systemen vollzogen werden. Im Gegensatz zu diesen werden im Büro nur eingeschränkte Funktionen benötigt, d.h., es sind keine aufwendigen Koordinatentransformationen, Spline-Funktionen und insbesondere kaum Operationen im dreidimensionalen Bereich erforderlich. Schon allein die verfügbare Peripherie verbietet einen Einsatz anspruchsvoller Grafikwerkzeuge, da die im Büro verfügbare Maus als Zeigeinstrument sowie die häufig anzutreffenden kombinierten Plot-/Druckgeräte ihren Pendanten im CAD-Bereich, Tablett und Hochleistungsplotter, in puncto Auflösung und Geschwindigkeit hoffnungslos unterlegen sind.

Die meisten üblichen Grafikeditoren orientieren sich an einem der beiden im folgenden beschriebenen Bearbeitungsmodellen; als beispielhaft können hierfür die Pakete *MacDraft* und *MacPaint* angesehen werden, die beide unter dem System *Macintosh* der Apple Corp. lauffähig sind.

- (i) *Edieren auf der Basis geometrischer Primitive:* Die Grafik wird als Agglomeration geometrischer Grundprimitive erzeugt und auch intern repräsentiert; meist stehen in diesem Fall komplexere Primitive (z.B. Rechteck, Oval etc.) zur Verfügung, so daß der Abstraktionsgrad oberhalb von Standards wie z.B. GKS (vgl. Ab-

schnitt 2.1.2) zu sehen ist. Zumeist werden auch zumindest primitive Segmentierungsmechanismen (GROUP/UNGROUP) unterstützt.

- (ii) *Edieren auf Basis des Paint-Ansatzes:* Ähnlich wie im vorstehenden Fall werden geometrische Primitive zur Verfügung gestellt, jedoch sofort nach Erzeugung auf der Zeichenfläche durch Rasterisierung in Bitmap-Darstellung übergeführt. Dadurch stehen für die Weiterbearbeitung grundsätzlich andere Funktionen zur Verfügung. MacPaint z.B. bietet hierfür unterschiedlich dimensionierte Schreib- und Löschwerkzeuge an (*Sprühdose*, *Radiergummi*) und erlaubt filigranes Arbeiten auf Pixelebene. Hingegen sind nach der Plazierung eines Primitivobjektes alle weiteren objektorientierten Bearbeitungsstrategien unmöglich. Die *on-the-spot* Rasterisierung verhindert die Ausgabe der Grafik auf hochauflösenden Ausgabegeräten bzw. die Weiterbearbeitung an Arbeitsplatzrechnern mit höherer Bildauflösung, da die Grafik sofort an die Rastereigenschaften des erstellenden Arbeitsplatzes gebunden wird.

Insbesondere aufgrund der letztgenannten Eigenheit von Paint-Editoren, die eine Geräteabhängigkeit impliziert und dadurch den Austausch von Grafik *de facto* unmöglich macht, ist im Bürobereich der kalligrafische Ansatz zu bevorzugen.

2.2.4 KOMMUNIKATIONSWERKZEUGE

Kommunikationswerkzeuge dienen dem Nachrichtenaustausch zwischen Partnern, die sich auf eine gemeinsame Basis im Hinblick auf die zugrundegelegten Datenstrukturen und Protokolle geeinigt haben.

Die entsprechenden Werkzeuge werden häufig als *Elektronische Post* bzw. *Electronic Mail Systems* bezeichnet. Ziel ihrer Einführung ist einerseits die Beschleunigung des Nachrichtenflusses durch automatische Übertragung und Verteilung auf elektronischen Medien, andererseits steht die Senkung der Kosten des Nachrichtenaustauschs im Mittelpunkt. Dieses Ziel wurde nun auch von den Trägern öffentlicher Datennetze aufgegriffen; entsprechende Normvorschläge wurden im Jahre 1984 durch die internationale Dachorganisation der Postbehörden (CCITT) veröffentlicht (vgl. SCHINDLER84, RACKE86).

Typisch für den Betrieb eines derartigen Kommunikationssystems ist die Einbettung in ein System mit Hintergrundrechner und Arbeitsplatzrechnern (z.B. Xerox Star; vgl. XEROX82 Kapitel 6); jedoch auch die Einbettung in Systeme mit Teilnehmer-/Teilhaberbetrieb mit Gateways zu benachbarten Systemen ist durchaus üblich (vgl. u.a. BEHRINGER85, DIGITAL85, SCHOLL85).

Zum Betrieb eines Kommunikationssystems müssen folgende Komponenten definiert sein:

- ein *Nachrichtenmodell*, das sowohl die festen Bestandteile einer Nachricht wie Absender, Adressat, Versandmodus sowie statistische Daten als auch die variablen Teile einer Nachricht definiert - dabei sind nicht nur alphanumerische Nachrichten, sondern auch vektorisierte oder rasterisierte grafische Information aufzugreifen (ein Plädoyer für Rasterübertragung im ostasiatischen Sprachraum aufgrund der Komplexität der alphabetischen Erfassung liefert SAKATA85, S. 115) - ,
- ein *Benutzermodell*, das die Ablage von versandten und empfangenen Nachrichten sowie deren Wiedergewinnung regelt, zudem aber auch für den Benutzer

- in der Rolle des *Senders* individuell oder global Versandarten und Verteiler,
- in der Rolle des *Empfängers* Filter zur Aussonderung unwichtiger Nachrichten (*junk mail*) oder Weiterleitung delegierter Aufgaben vorsieht.

Während rudimentäre Systeme auf der Basis primitiver Dokumentenmodelle und Versandarten derzeit auf fast allen Mehrplatz- bzw. netzgekoppelten Systemen zur Verfügung stehen, sind neben der Weiterentwicklung auf dem Gebiet der multimedialen Kommunikation (siehe Abschnitt 2.3) noch folgende weiteren Entwicklungsrichtungen erkennbar:

- *Adressierungsschemata*: Ein Adressierungsschema umfaßt die Sammlung von grundsätzlichen Annahmen über die Empfänger innerhalb eines Kommunikationssystems und die formalen Vorschriften zu ihrer Beschreibung. TSICHRITZIS⁸⁴ unterscheidet dabei als Grundtypen das *Absenderschema*, das die volle Beschreibung der Empfänger durch den Sender verlangt, das *Weiterleitungsschema*, das aufgrund lokaler Weiterleitungsstrategien operiert, und das *Empfängerschema*, bei dem keine eigentliche Adressierung erfolgt, sondern der Empfänger selbst für die Erkennung der Relevanz einer Nachricht verantwortlich ist. Ziel der Betrachtungen ist, das derzeit in den meisten Systemen vorherrschende Absenderschema durch Weiterleitungs- bzw. Filterstrategien in die Lage zu versetzen, bloße Kennzeichnungen des Empfängerkreises in Empfängeridentifikationen abzubilden, ohne gravierende Verluste an Relevanz für den Empfänger hinnehmen zu müssen. In jedem Fall muß dabei die *Vollständigkeit* des Adressierungsschemas, d.h. die Gewährleistung des Eintreffens der Nachricht bei einem qualifizierten Empfänger, erhalten bleiben (vgl. hierzu MARTIN⁸⁶).

- *Intelligente Nachrichten:* Eng assoziiert an die Arbeiten zum Thema *Adressierungsschemata* sind die Entwicklungen auf dem Gebiet der *Intelligenten Nachrichten* (vgl. hierzu HOGG83, HOGG85a). Hierbei werden im Gegensatz zur herkömmlichen elektronischen Kommunikation - der in diesem Kontext benutzte Terminus Technicus lautet *email (electronic mail)* - keine passiven Nachrichten an einen festen Empfängerkreis übertragen, sondern vielmehr Programme (*scripts*), die somit *aktive* bzw. *intelligente* Nachrichten (*imail*) instantiieren. Nach der Übertragung an einen Initial-Empfängerkreis wird aufgrund eines Dialogs mit dem jeweiligen Empfänger einerseits die von ihm benötigte Information eingeholt, andererseits aber auch aufgrund der Antworten sowie im Rahmen des Scripts formulierten Wissens *autark* über das weitere Schicksal der Nachricht (Rückkehr zum Ausgangspunkt, Weiterver- sand an andere Empfänger) entschieden. So lassen sich z.B. gezielte Umfragen starten, die dann mit den gesam- melten Informationen an den Ausgangspunkt zurückkehren und dort statistisch ausgewertet werden können. Wei- terhin sind auf der Basis dieses Konzepts auch flexible Adressierungsschemata realisierbar.
- *Dokumentanalyse:* Einen weiteren Ansatz zur nachträglichen Präzisierung von Adressierungen bietet die Dokumentenanalyse und -deskribierung nach formalen und inhaltlichen Kriterien. Dies ist im Falle der Über- mittlung durch öffentliche Dienste (z.B. Teletex) von besonderer Bedeutung, da in diesem Fall nur in seltenen Fällen ein Schriftstück den Adressaten direkt erreicht, sondern meist eine Mittlerstelle (Posteingang) ein- greifen muß, um Zuständigkeiten festzustellen und die Adresse dahingehend zu präzisieren. Einen Überblick über die dabei angewendeten Techniken sowie die Normie- rungsaktivitäten zur Erleichterung des zwischen-

betrieblichen Schriftverkehrs bietet SCHUMANN⁸⁵.

Während meist der Nachrichtenaustausch *asynchron*, d.h. ohne Warten auf Rückmeldung durch den Empfänger, erfolgt, stehen in manchen Fällen auch Werkzeuge zur *synchronen* Kommunikation zur Verfügung, deren Übertragungsprotokoll dem der fernmündlichen Kommunikation entspricht. Derartige Systeme sind jedoch nur bei Einsatz sehr schneller Kommunikationsmedien (LAN) verwendbar.

2.2.5 DECISION-SUPPORT-SYSTEME

Decision-Support-Systeme (DSS) sind auf einem funktionalen Entscheidungsmodell basierende Dialogsysteme und dienen der Vorbereitung von Entscheidungen auf der dispositiven und strategischen Ebene der betrieblichen Organisation. Das Ziel ist die *Unterstützung* (im Gegensatz zu einer *Automatisierung*) des Entscheidungsprozesses. Weitere Merkmale eines DSS sind (nach MERTENS⁸³):

- *Herrensystem*: Ein DSS wird im Mensch-Computer-Dialog benutzt, d.h., die Initiative in der Dialogführung liegt beim Menschen.
- *Parametrierbarkeit & Adaptivität*: Um auf nicht a priori vorgegebene Problemstrukturen eingehen zu können, müssen Änderungen bestehender Modelle leicht möglich sein; zu diesem Zweck muß eine Programmierung auf sehr hohem Abstraktionsniveau ermöglicht werden, die es selbst Endbenutzern gestattet, korrektiv einzugreifen. Der Kompatibilität mit anderen Werkzeugen sowie der optimalen Ausnutzung von Rechnerressourcen

wird dabei nur sekundäre Bedeutung beigemessen.

- *Infrastruktur*: Die Umgebung eines DSS sollte Lösungen für numerische Probleme (Finanzmathematik, Statistik), Aufbereitung und Ausgabe (Bürografik) sowie Nutzwert- oder Risikoanalyseverfahren in Methodenbanken anbieten. In den meisten Fällen stellen die Verfahren und Modellierungsansätze des *Operations Research* die Basis für die einzusetzenden Methoden dar, wobei das Hauptgewicht auf diskreten, weniger auf linearen Modellen liegt.

Datenbankgestützte DSS stellen im Metadatenbereich (Data Dictionary) sehr hohe Anforderungen, da sie aufgrund der Verschiedenartigkeit der Datenorganisation der relevanten Daten nicht nur Datenbanksysteme, sondern auch herkömmliche Dateiorganisationen behandeln können müssen.

Der Begriff *Decision-Support-Systeme*, wie er bisher in diesem Abschnitt erläutert wurde, bezieht sich auf eine Werkzeuggruppe, die in der Literatur als *DSS-Generatoren* oder *Generische DSS* bezeichnet wird (vgl. hierzu ARIAV85, S. 1049). Häufig finden sich in den kommerziell angebotenen Büroinformationssystemen auch Komponenten, die spezielle Probleme behandeln, vergleichsweise gut in die übrigen Werkzeuge und deren Datenbasen integriert sind, dafür jedoch dem Benutzer relativ geringe Erweiterungsmöglichkeiten lassen. Typische Einsatzgebiete für derartige Komponenten sind Projektplanung nach einem der gängigen Verfahren (CPM, MPM, PERT) oder Führen von Terminkalendern (siehe z.B. Terminplanung in CEO, SCHOLL85, S. 76).

2.3 MULTIMEDIALE DATENVERWALTUNG

Der Begriff *Multimediale Datenverwaltung* stützt sich nicht auf den Begriff des physischen Mediums im Sinne von Trägersubstanz für Information, sondern bezeichnet als Medium eine Präsentationsfläche, auf der Information in diversen Arten organisiert sein kann. Beispiele hierfür wurden als Basisdatentypen in Abschnitt 2.1 eingeführt. Derzeit sind auf dem Markt diverse Systeme verfügbar, die innerhalb einer Kompatibilitätssphäre mehrere Werkzeuge im Sinne der in Abschnitt 2.2 angeführten Basistypen anbieten (OPEN ACCESS, LOTUS 1-2-3 etc.). Entscheidend für den Grad der Kompatibilität ist dabei die Menge der Bearbeitungsfunktionen, die im Rahmen der Gesamtbearbeitung eines Dokuments zur Bearbeitung des Einzel-Datentyps zur Verfügung bleiben, sowie ein Maß für die Erhaltung des Abstraktionsgrades des Organisationsmodells.

Der im folgenden häufig benutzte Begriff *Organisationsmodell* legt die Begriffe fest, die zur Beschreibung der Logischen Struktur und der Externen Repräsentation Anwendung finden. Auf diese Weise wird auch der Gebrauch der Funktionen normiert, die diese Strukturen manipulieren bzw. interpretieren. Letztendlich wird dadurch eine Sammlung definierter Zugriffs- und Manipulationsschnittstellen ähnlich wie bei einem *Datenmodul* verstanden, gleichzeitig aber auch die Aufteilung und Gestaltungsmittel eines *dedizierten* oder *virtuell-abstrakten* Bildschirmgerätes. COUTAZ84 und COUTAZ85 bauen eine derartige Abstraktion schrittweise auf: von der

- *Logischen Geräteabstraktion*, die die Funktionen *physischer* Geräte auf die Funktionen *imaginärer* logischer Geräte abbildet (vgl. hierzu u.a. das Geräteabstraktionskonzept in GKS, das in ENDERLE83 beschrieben ist) und so *Geräteunabhängigkeit* der Applikationsprogramme bewirkt, über die

- *Virtuelle Datenendgerät abstraction*, durch die auf ein physisches Datenendgerät Ausprägungen einer Reihe ggf. verschiedener virtueller Datenendgeräte abgebildet werden, wobei diese außer der einfachen Emulation von Standardgeräten wie VT 220 oder Tektronix 4104 auch bereits Applikationsspezifika einbeziehen und als
- *Objektorientierte Ein-/Ausgabeabstraktion* auf die direkte Manipulation der behandelten Datenstrukturen abzielt.

Als Beispiel für ein Organisationsmodell sind im Falle der Tabellenkalkulation die Begriffe *Zelle* (einschließlich dem zugehörigen Adressierungsschema), *Konstante* und *Formel* bzw. *Regel* von Bedeutung; als Funktionen sind die in Abschnitt 2.2 skizzierten Edierfunktionen sowie ein Verfahren zur Neuberechnung aller abhängigen Werte einzuführen. Der Begriff *Organisationsmodell* überdeckt den Begriff *Subarchitektur* (*Content Architecture*), der in den Standardisierungen von ECMA Verwendung findet (vgl. Abschnitt 2.3.2).

Der Abstraktionsgrad eines Organisationsmodells definiert eine Halbordnung auf der Menge dieser Modelle. Ein Modell B ist *konstruktiv-abhängig* von einem Modell A, wenn alle Funktionen auf Elementen des Modells B bzw. die Externe Repräsentation dieser Elemente selbst auf Elemente und Repräsentationen des Modells A rückführbar sind. So ist z.B. ein Organisationsmodell auf Basis der Tabellenkalkulation auf Elemente der Basisgrafik einerseits und der Textverarbeitung andererseits rückführbar, diese beiden wiederum basieren letztendlich auf der Rastergrafik als Grundmodell.

In den weiteren Ausführungen wird auch der Begriff der Kompatibilität bzw. Verträglichkeit häufiger eine Rolle spielen. Die vielschichtige Bedeutung des in der Literatur relativ unkritisch eingesetzten Begriffs kann zunächst in folgende Aspekte aufgeteilt werden:

- *Ablaufkompatibilität:* Hierunter versteht man die Verträglichkeit mehrerer Ablaufumgebungen (z.B. Betriebssysteme) in bezug auf bestimmte Klassen von Funktionen (Programme).
- *Kompatibilität bzgl. der Internen Repräsentation:* Sie bezeichnet die Verträglichkeit von Internen Repräsentationen abstrakter Datentypen bei Verarbeitung durch verschiedene Werkzeuge; damit wird der Bereich der Austauschformate abgedeckt (*IR-Kompatibilität*).
- *Kompatibilität bzgl. der Externen Repräsentation:* Sie bezeichnet die Integrationsfähigkeit von Werkzeugen hinsichtlich ihrer Externen Repräsentationen (*ER-Kompatibilität*). Der Fall der Kompatibilität auf der Basis der zugrundeliegenden Logischen Struktur ist hierin eingeschlossen.

Bei letzterer wiederum kann man folgende Kategorien unterscheiden:

- *Repräsentative Kompatibilität:* Darunter versteht man die Fähigkeit eines Werkzeugs, die Externe Repräsentation von Objekten aufzunehmen, die durch andere Werkzeuge erzeugt wurden. Dabei lassen die Werkzeuge jedoch keine Weiterverarbeitung auf der Basis des Ursprungs-Organisationsmodells der übernommenen Objekte zu.

- *Funktional-reduzierte Kompatibilität:* Dies bezeichnet die Fähigkeit eines Werkzeugs, die Externe Repräsentation von Objekten aufzunehmen, die durch andere Werkzeuge erzeugt wurden. Eine Weiterverarbeitung der übernommenen Objekte auf der Basis eines Organisationsmodells mit verringertem Abstraktionsgrad ist möglich.
- *Funktionale Kompatibilität:* Es kann dabei von einer eigentlichen Übernahme einer Externen Repräsentation keine Rede sein: alle relevanten Werkzeuge arbeiten konvivial auf einer gemeinsamen Darstellungsfläche.

2.3.1 KONZEPT EINER BÜROORIENTIERTEN DOKUMENTENARCHITEKTUR

Anforderungen. Der in dieser Abhandlung eingeführte Dokumentbegriff fordert die Generalisierung des Begriffsinhaltes, der durch die einzelnen isolierten Basiswerkzeuge implizit eingeführt wurde. Durch die Forderung funktionaler ER-Kompatibilität innerhalb der Menge der einsetzbaren Werkzeuge wird die Darstellung von Information mit Hilfe verschiedener Medien innerhalb eines Kontexts möglich. Analoge Begriffe wie *kanonischer Aufbau* und *Automat* benutzt DÜRRE83 in der Beschreibung der Architektur eines Systems mit ähnlicher Zielsetzung. Diese Anforderungen gelten in allen Bereichen, für die ein multimedialer Dokumentbegriff erforderlich ist, also neben der Büroautomation auch für Dokumentation sowie im technischen Bereich; eine auf die letzteren Bereiche bezogene Beschreibung eines derartigen Systems ist in KRATZER86 enthalten.

Koexistenz von Datentypen und Bearbeitungswerkzeugen. Der Begriff *Dokument* wird in Intension und Verwendung durch die folgenden Benutzungsregeln präzisiert:

- (i) Ein Dokument ist eine durch ein *endliches Koordinatensystem* aufgespannte *rechteckige Darstellungsfläche*.
- (ii) Die Achsen entsprechen in Bemaßung und Bedeutung dem für die Darstellungsfläche gewählten Organisationsmodell.
- (iii) Dabei steht eine Transformation in ein *normalisiertes Koordinatensystem* als Basis zur Geräteein-/ausgabe zur Verfügung.
- (iv) Ein Dokument ist die Wurzel einer Hierarchie von *Rahmen* (*frames*), innerhalb derer unterschiedlich typisierte Teile des Dokuments, im folgenden auch als *Fragmente* bezeichnet, bearbeitet werden.
- (v) Rahmen sind rechteckige Darstellungsflächen und einem einheitlichen Organisationsmodell unterstellt, mit dessen Hilfe ein Objekt repräsentiert wird; das Organisationsmodell als Summe von Manipulations- und Datenverwaltungsfunktionen wird durch ein *Werkzeug* implementiert, das jedem Rahmen eindeutig zugeordnet ist.
- (vi) Alle in der Hierarchie einem Rahmen unterstellten Rahmen sind in diesem voll enthalten.
- (vii) Rahmen sind hinsichtlich ihrer *materiellen Eigenschaften* (transparent, undurchsichtig), ihrer *Teilbarkeit*, ihrer *Aufnahmefähigkeit* für unter- oder beigeordnete Rahmen und ihrer eigenen Verschieb- und Vergrößerbarkeit und der untergeordneter Rahmen attri-

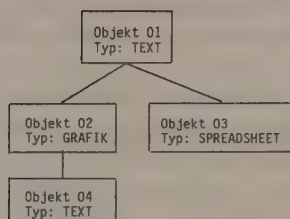
butierbar, d.h., diese Eigenschaften können durch Parameter festgelegt werden.

Analog hierzu gelten Regeln für das Protokoll, das die Werkzeuge untereinander beachten müssen:

- (i) Jedes in einem Rahmen aktive Werkzeug muß sich in seiner Darstellung auf die normalisierte Abmessung des umgebenden Rahmens beschränken.
- (ii) Jedes dieser Werkzeuge muß die Einbettung anderer Rahmen berücksichtigen, sofern eine derartige Einbettung aufgrund der Rahmenattributierung möglich ist.
- (iii) Die Benutzerschnittstelle des Werkzeugs muß in der Lage sein, sich flexibel den jeweiligen Rahmengrenzen anzupassen; ideal wäre ein Einsatz abstrakter Operatoren unter Verwendung eines gemeinsamen Dialog-Verwaltungsprogramms zur Vereinheitlichung der Schnittstelle.
- (iv) Eventuelle Layout-Operationen (Verschiebung, Größenveränderung, Teilung) innerhalb des für ihn gültigen Rahmens bleiben dem verantwortlichen Werkzeug vorbehalten; die gilt insbesondere für Rahmen zu Textbearbeitungswerkzeugen.

Abbildung 2-3 zeigt ein Beispiel für die Logische Struktur und Externe Repräsentation eines Dokuments nach diesen Konventionen.

Logische Struktur



Externe Repräsentation

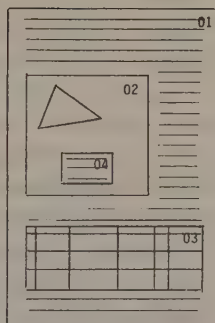


Abb. 2-3: Beispiel für Logische Struktur und Externe
Repräsentation eines Dokuments

Implementierung des Dokumenten-Organisationsmodells. Die Kontrollinstanz, die den Bearbeitungswerkzeugen übergeordnet ist, ist selbst wiederum als Implementierung eines Organisationsmodells anzusehen, das sich in diesem Fall auf die Verwaltung von Rahmen und die Verteilung der Kontrolle an untergeordnete Instanzen beschränkt. Demzufolge bearbeitet diese Instanz, im folgenden *Dokumentmonitor (DM)* genannt, eine strukturelle Abstraktion des Dokuments, während die untergeordneten Werkzeuge die inhaltliche Bearbeitung vornehmen. Die Aufgaben des Dokumentmonitors sind im einzelnen:

- o *Rahmenverwaltung*: Der DM legt Rahmen an und platziert sie in einem lokal auf den übergeordneten Rahmen bezogenen Koordinatensystem. Die dabei zu erfolgende Auswahl des bearbeitenden Werkzeugs und damit des lokal aufzubauenden Organisationsmodells impliziert die Rahmenattribute. Zur Unterstützung dieses Mechanismus müssen Informationen über Werkzeuge und ihre Anforderungen und Aufrufmodalitäten zur Verfügung stehen.
- o *Kontrollfluß*: Nach Aufbau bzw. Selektion des Rahmens wird das entsprechende Werkzeug aktiviert. Die Ein-/Ausgabeoperationen müssen dabei die Topologie der Dokumentseite beachten; in der Praxis heißt dies, daß die herkömmlichen Ein-/Ausgabeoperationen durch DM-spezifische Operationen ersetzt werden müssen, um den Zugriff auf Strukturinformationen zu ermöglichen.
- o *Strukturinformationen & Interne Repräsentation*: Der DM verwaltet Informationen über die Identität jedes Dokuments mit entsprechender Attributierung sowie über Attribute und Platzierung der konstituierenden Rahmen. Des weiteren sind zu jedem Dokument die Pfade zu den jeweiligen Internen Repräsentationen der Fragmente bekannt, so daß für Sicherungs- und Übergabezwecke jeweils der gesamte Kontext eines Dokuments bereitsteht. Im Grunde handelt es sich bei dem Übergang vom Dateibegriff zum Dokumentbegriff um einen Abstraktionsschritt, der mit dem Übergang auf die Bearbeitung mit Datenbanksystemen vergleichbar ist. In beiden Fällen wird von Systemaspekten abstrahiert, um damit mehr Freiheitsgrade für die Abspeicherung zu gewinnen, bzw., im Falle der Dokumentenverarbeitung, um auf eine explizite Formulierung einer Konvention hinsichtlich der Internen Repräsentation verzichten zu können.

- o *Invertierungs- und Suchdienste*: TSICHRITZIS85a nennt folgende Wege zur Wiedergewinnung bzw. Restaurierung von Dokumenten:
 - (i) *Beschriftung* in Form von Titelzeilen,
 - (ii) *Sequentielle Suche*,
 - (iii) *Abstraktion dokumentinhärenter Attribute*,
 - (iv) *Mustersuche* (inhaltsbezogen),
 - (v) *Ablage in Miniaturen*.

In einer Synthese aus den Ansätzen (i) - (iv) schlägt KRATZER86 für das DM-Konzept einen Ansatz vor, der sich an einem eingeschränkten Prädikatenkalkül orientiert, dabei jedoch die Struktureigenschaften von Dokumenten berücksichtigt. Die Invertierung von Dokumentfragmenten nach Abschluß eines Arbeitsschrittes, d.h. einer Ediersitzung, erfolgt typspezifisch; gleichermaßen sind auch die Abfragen mit Hilfe typspezifischer Prädikate, die in der Datenbasis von DM vordefiniert sein müssen, zu formulieren. Die Ergebnismengen nach einer Anfrage unterliegen einer weiteren Verfeinerung durch sequentielle Prüfung und Auswahl.

Dokumentnetze (Document Webs). Gemeinhin sind Dokumente durch eine Vielzahl von Querverweisen auf andere Dokumente charakterisiert. Dies entspricht der Art des Menschen zu denken, zu argumentieren und zu rasonieren. Der Gedanke, Dokumente nicht als individuell zu betrachtende Einheiten aufzufassen, sondern nur als Partikel in einer Gedanken- und Argumentationskette, wurde von V. Bush bereits im Jahre 1945 formuliert (vgl. MEYROWITZ82a, S. 399) und als *Hypertext-* bzw. *Hyperdokumentsysteme* in einer

Reihe von Implementierungen konkretisiert (vgl. dazu auch YANKELOVICH85). Ein Hyperdokument mag als Konglomerat von Einzeldokumenten entstanden sein, durch die Verflechtung von Dokumentfragmenten durch Ausprägungen von Beziehungstypen untereinander wird dieser Eindruck wieder verwischt und dem Leser die Möglichkeit geboten, seinen eigenen Interessen statt dem vom Autor vorgegeben Informationsfluß zu folgen, wie dies visionär in DISESSA86 bezüglich der Boxer-Dialog- und -Programmierungsumgebung dargestellt ist. Letztendlich stellen Hyperdokumente nur die Generalisierung des Gedankens des Aufbaus einer Dokumentenarchitektur durch eine Objekthierarchie dar: zu der ein Dokument konstituierenden Objektrelation enthält können noch weitere Relationen definiert werden. Diese Generalisierung führt dann auch zum Übergang auf die in diesem Feld eingebürgerte graphentheoretische Terminologie, die von *Knoten* statt *Objekten* spricht.

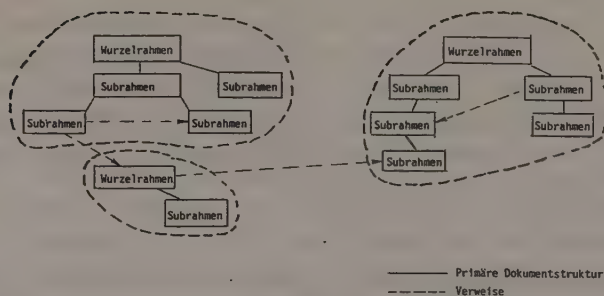


Abb. 2-4: Beispiel eines Hyperdokuments

Hyperdokumentsysteme zeichnen sich nach DELISLE86 durch folgende Eigenschaften aus:

- o *Verwaltung*: Neben der Verwaltung von multimedialer Information in einem Ediersystem sind der Aufbau und die Verwaltung von Verbindungen zwischen einzelnen Dokumentfragmenten möglich. Diese Verbindungen sind namensbehaftet und erlauben so die Erzeugung von Sub-Hyperdokumenten durch Bildung der transitiven Hülle über Verbindungen mit bestimmten Beschriftungen. Falls die Erzeugung von Versionen auf Knotenebene vorgesehen ist, müssen analoge Mechanismen auch für Beziehungen zwischen Knoten vorhanden sein.
- o *Traversierung*: Die Auswertung und Verfolgung eines derartigen Dokuments ist nur Hilfe eines dynamischen Mediums möglich, wie es ein Sichtgerät bzw. ein Arbeitsplatzrechner darstellt. Die Relation enthält nimmt dabei eine hervorragende Stellung ein, da sie neben der bloßen Bezugnahme auf ein untergeordnetes Objekt auch Layout-Aspekte des Dokumentes berücksichtigen muß. Die Auswertung weiterer Verbindungen wird dann durch Aufblenden der referenzierten Objekte ermöglicht. Gleichzeitig muß durch ein Linearisierungsinstrument (wie im Falle des Systems Neptune, siehe DELISLE86) eine Möglichkeit der Ausgabe auf Papier oder ähnlich linear organisierte Medien geschaffen werden.
- o *Benutzerschnittstelle*: Neben den bereits oben erwähnten Funktionen der Dokumentverwaltung müssen auch Orientierungshilfen nebst einem Sicht-Mechanismus zur Verfügung stehen, die der Navigation in einem Hyperdokument dienen. Hilfreich hierfür ist ein Sicht-Mechanismus, der bestimmte Bereiche des Dokuments durch Beschränkung auf bestimmte Verbindungstypen ausblendet.

Der Dokumentmonitor realisiert die Erzeugung von Hyperdokument-Strukturen durch Generierung von Verbindungen auf Rahmenebene. Die verfügbaren - und mit der zu einem gegebenen Zeitpunkt gültigen Sicht-Beschreibung konformen - Verbindungen werden zur Auswahl gestellt und ggf. weiterverfolgt. Eine Protokollkomponente erlaubt die Zusammenstellung eines individuellen Pfades durch das Hyperdokument. Mit Hilfe des Pfades kann dann nach entsprechender Linearisierung ein den Benutzerwünschen entsprechender Ausschnitt des Hyperdokuments erstellt werden.

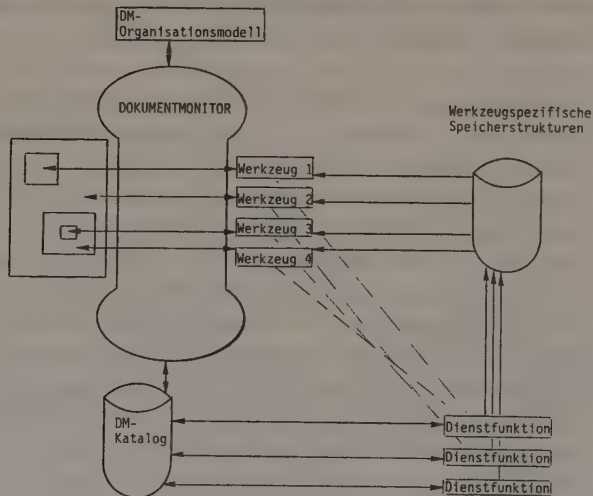


Abb. 2-5: Übersicht über die Architektur des Dokumentmonitors

Der Dokumentmonitor - Architektur und Datenstrukturen. Abb. 2-5 zeigt eine Übersicht über die Architektur des

Dokumentmonitors. Während die integrierende Funktion des DM durch Vermittlung des Kontrollflusses zwischen den einzelnen beteiligten Werkzeugen bereits beschrieben wurde, sollen nun im einzelnen die Datenstrukturen des DM-Katalogs aufgeführt und kommentiert werden.

Der DM-Katalog versteht sich als Sammlung aller Informationen über Dokumentstrukturen und Bearbeitungswerkzeuge. Zugleich verwaltet er Prozeduren, die der Ablage, Invertierung nach bestimmten Kriterien sowie der Wiederauffindung von Dokumenten dienen. Abb. 2-6 zeigt eine Übersicht über die erforderliche Datenstruktur in Form eines Konstruktionsdiagramms (*).

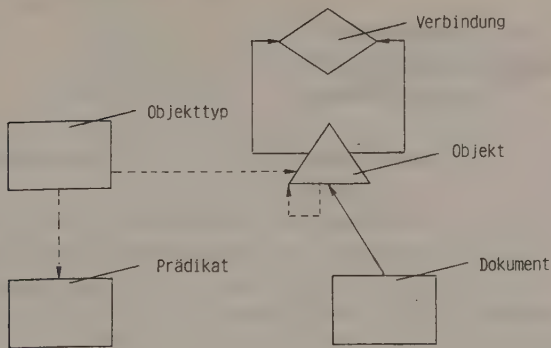


Abb. 2-6: DM-Katalog - Konstruktionsdiagramm

(*) Das Diagramm folgt der Notation für Entity-Relationship-Diagramme nach WEDEKIND81 (S. 124f); die Beschreibung benutzt eine quasi-relationale Notation, ist jedoch in andere Schreibweisen überführbar, sofern Komplexbildung und anschließende Selektion vorausgesetzt werden können.

Im einzelnen bestehen die Datenstrukturen aus:

Schema:

objekttyp (*ot#*, *name*, *werkzeug*, *dienstprogramm*)

<i>ot#</i>	Eindeutige Kennzeichnung eines Objekttyps
<i>name</i>	Name des Objekttyps (mnemonische Hilfestellung)
<i>werkzeug</i>	Bezugnahme auf das Bearbeitungswerkzeug, das die Summe der für den Objekttyp definierten Funktionen enthält
<i>dienstprogramm</i>	Bezugnahme auf ein Prozedurbündel, das folgende Dienste leistet:

- Invertierung der Internen Repräsentation nach dem vorliegenden Typ
- Auswertung von Prädikaten, die für den vorliegenden Typ definiert sind (die jeweiligen Parameter werden als Liste mitübergaben)

Schema:

objekt (*o#*, *superobjekt*, *plazierung*, *typ*,
attribute, *zugriff*)

<i>o#</i>	Eindeutige Kennzeichnung eines Objekts, das extern als Rahmen repräsentiert wird
<i>superobjekt</i>	Bezugnahme auf das hierarchisch übergeordnete Objekt hinsichtlich der <i>enthält</i> -Relation (Wertebereich <i>o#</i>)
<i>plazierung</i>	Plazierung der linken oberen Rahmenecke im Normalkoordinatensystem des übergeordneten Rahmens
<i>typ</i>	Bezugnahme auf den Objekttyp, dem das Objekt angehört (Wertebereich <i>ot#</i>)
<i>attribute</i>	Beschreibung der gültigen Rahmenattribute
<i>zugriff</i>	Beschreibung des werkzeugspezifischen Zugriffspfads (system- und werkzeugabhängig)

Schema:

dokument (*d#*, *dokument-info*)

<i>d#</i>	Eindeutige Kennzeichnung eines Dokuments; die Nummer entspricht dabei den <i>o#</i> der Objekte, die kein hierarchisch übergeordnetes Objekt hinsichtlich der <i>enthält</i> -Relation haben
<i>dokument-info</i>	Eine Sammlung von administrativen Daten zum Dokument; Beispiele für auftretende Attribute wären:

- Autor
- Historie der Erstellung und Modifikation
- Zugriffsrechte

- Deskriptoren

Schema:

verbindung (*von-objekt*, *zu-objekt*, *name*)

Die Datenstruktur *verbindung* ist in diesem Fall für eine gerichtete Beziehung ausgelegt (*referenziert*, *ist-konstruktiv-abgeleitet-aus*); im Fall der ungerichteten Beziehung (*ist-sachlich-zusammenhängend-mit*) ist die Reihenfolge der angesprochenen Objekte irrelevant.

<i>von-objekt</i>	Eindeutige Kennzeichnung des Subjekts hinsichtlich der Verbindung (Wertebereich 0#)
<i>zu-objekt</i>	Eindeutige Kennzeichnung des Objekts hinsichtlich der Verbindung (Wertebereich 0#)
<i>name</i>	Name der Verbindung

Schema:

prädikat (*typ*, *name*)

<i>typ</i>	Eindeutige Kennzeichnung des Objekttyps, für den das Prädikat definiert ist (Wertebereich 0t#)
<i>name</i>	Name des Prädikats

2.3.2 NORMIERUNGSBESTREBUNGEN

Grundlegende Betrachtungen. Dokumente sind seit langem Gegenstand ausgedehnter Normierungsbestrebungen in verschiedenen nationalen und internationalen Gremien der einschlägigen Institutionen. Im Gegensatz zu dem im vorhergehenden Abschnitt vorgestellten Modell, das aus einer Synthese von Werkzeugen mit Hilfe eines Monitors ein übergeordnetes Organisationsmodell implementierte, sind diese Normen *werkzeugunabhängig* und damit aufgefordert, eine volle Spezifikation auch der Rahmeninhalte mitzuliefern, die in DM in den Bereich der Verantwortung der einzelnen Werkzeuge fällt.

Neben herstellerinternen und damit nur begrenzt einsetzbaren Standards wie DCA von IBM (vgl. DESOUSA81 bzw. GMELCH86) oder Interscript von Xerox konzentrieren sich die Bemühungen um die Standards für eine *Office Document Architecture (ODA)* und ein entsprechendes Austauschformat *Office Document Interchange Format (ODIF)* (*). Die Standards wurden zunächst maßgeblich von ECMA (*European Computer Manufacturers Association*) beeinflusst, später jedoch von ISO aufgegriffen, nachdem sich dort die Arbeiten anfangs vornehmlich auf den Ansatz des Dokumentaustauschs über Markup-Sprachen (*GSML*) konzentriert hatten (vgl. SCHELLER86). Die Arbeiten im ODA/ODIF-Bereich berücksichtigen dabei die bereits bestehenden Normen im Bereich des Nachrichtenaustauschs der CCITT (X.400ff-Protokolle, dabei in erster

(*) Weitere Arbeiten in diesem Umfeld zielen auf die Ergänzung des ODA/ODIF-Standards um eine konzeptionelle Beschreibungskomponente ab; der dabei erstellte Vorschlag einer CSD (*Conceptual Structure Definition*) soll unabhängig von der Architektur der Einzeldokumente inhaltsbezogene Abfragen ermöglichen. Details zu CSD sind u.a. in GMELCH86 (S. 44ff) zu finden.

Linie X.409: *Presentation Transfer Syntax and Notation* und X.420: *Interpersonal Message User Agent Layer*; vgl. auch SCHINDLER84 und RACKE86).

Grundprinzipien von ODA. Die folgenden Ausführungen beziehen sich auf den richtungsweisenden Dokumentenarchitekturansatz von ECMA (ECMA 101). Vertiefende Informationen finden sich in ELUMENFELD85, SCHÜRR85 und HORAK85b.

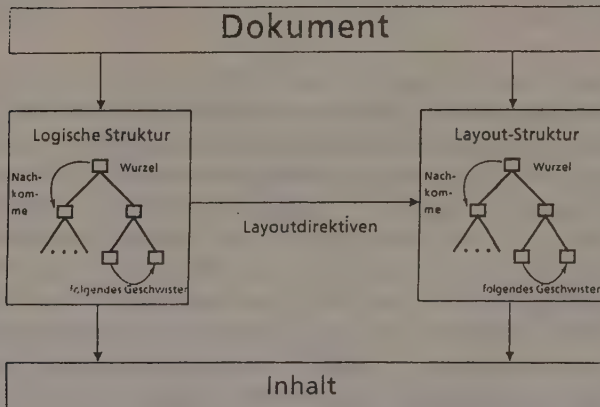


Abb. 2-7: Die Dualität der logischen Struktur und der Layout-Struktur (aus SCHÜRR85, S. 6)

ODA ist grundlegend geprägt durch die Dualität der logischen Struktur von Dokumenten einerseits und der physischen Gestalt andererseits (vgl. Abb. 2-7). Daraus wird ersichtlich, daß ein

Dokument über eine logische Struktur hinaus gemäß explizit formulierter Layout-Direktiven auch über eine Layout-Struktur verfügt. Die logische Struktur wird über eine Hierarchie logischer Objekte beschrieben, die im einzelnen auf folgenden Ebenen anzusiedeln sind:

- *Dokument*,
- eine Hierarchie zusammengesetzter logischer Objekte, die durch die Konstruktoren Sequenz, Auswahl und Agglomeration aus Objekten niedrigeren Niveaus zusammengesetzt werden,
- als Blattknoten schließlich logische Basisobjekte, die nach dem derzeitigen Stand der Normierung entweder durch Subarchitekturen vom Typ *Text* oder vom Typ *Rastergrafik* sind. Eine entsprechende Normung für Vektorgrafik auf der Basis des GKS-Standards ist in Vorbereitung,

oder formal (in BNF-Notation)

```
(logical) document ::= (composite | basic)*
composite          ::= (composite | basic)*
basic.
```

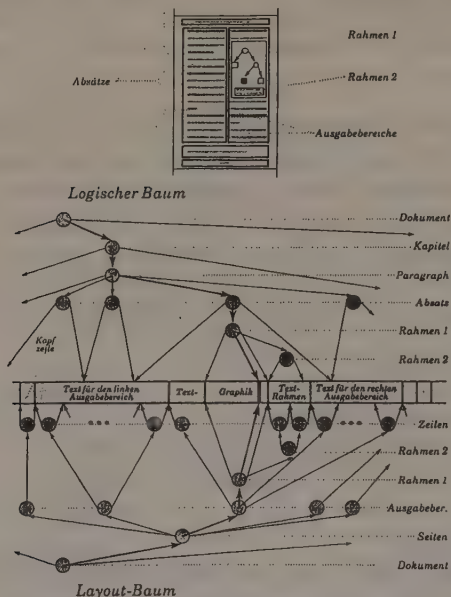
Die physische Struktur baut auf folgenden Elementen auf:

- *Dokument*,
- einer Hierarchie von *Seitenfolgen*,
- einer *Seite* entweder als atomarem oder zusammengesetztem Objekt
- im letzteren Fall noch dem *Block* als physischer Repräsentation eines Rahmens (ggf. nach Teilung),

oder formal (in BNF-Notation)

```
(layout) document ::= (pageset | page)*
pageset            ::= (pageset | page)*
page               ::= (frame | block)
frame              ::= (frame | block)
block.
```

Ein Beispiel für die Abbildung der logischen Struktur in die Layout-Struktur unter Einbezug grafischer Elemente zeigt Abb. 2-8.



Ausschnitt einer Dokumentenseite aus der internen Datenstruktur eines Dokuments.
Für das selektierte graphische Objekt im Graphik-Rahmen (schwarzes Quadrat) ist der
Selektionspfad im Doppel-Baum durch dicke Pfeile markiert.

Abb. 2-8: Ausschnitt der ODA-Strukturen für eine Dokumentseite (aus SEIGIS86, S. 24)

Neben der Definition spezifischer Dokumente ist auch die Definition von logischen und Layout-Dokumentklassen möglich, wobei Dokumentklassen sich aus Einzelobjekten zusammensetzen, die wiederum durch verschiedenartige Typisierung durch Attributie-

nung charakterisiert sind. Dadurch ist es z.B. möglich, fortlaufende Kapitelnummern und ein einheitliches Erscheinungsbild der Numerierung dadurch zu erzwingen, daß *Kapitelnummer* als Attribut mit entsprechenden Rechenregeln einer Klasse *Kapitel* angefügt wird. Derartige Dokumentklassendefinitionen sind gleichfalls Träger der benötigten Information zur Interpretation und Weiterverarbeitung eines Dokuments. Es wird so versucht, eine *werkzeuginvariante* Bearbeitung des Dokuments zu sichern. Den Gesamtaufbau eines Dokuments unter Einschluß der Klassendefinition zeigt Abb. 2-9.

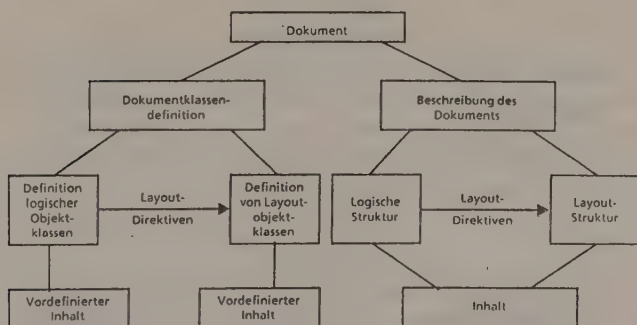


Abb. 2-9: Architektur eines typisierten Dokuments
(aus HORAK85a, S. 63)

Dokumentaustausch. Das Austauschformat nach der ODIF-Norm besteht aus einer Folge von sog. *Deskriptoren* und *Texteinheiten*. Dabei enthält ein Deskriptor die Attribute, die einem Dokumentprofil, einem generischen oder einem spezifischen Objekt zugeordnet werden können. Texteinheiten wiederum repräsentieren

Inhalte und enthalten Attribute und Textelemente eines Inhaltsstückes. Abb. 2-10 zeigt den Aufbau des Datenstroms für den Dokumentenaustausch; der allgemeinen Dokumentbeschreibung folgt dabei der generische Teil mit der Dokumenttypenbeschreibung und anschließend der spezifische Teil mit der Ausprägung des konkreten Dokuments.

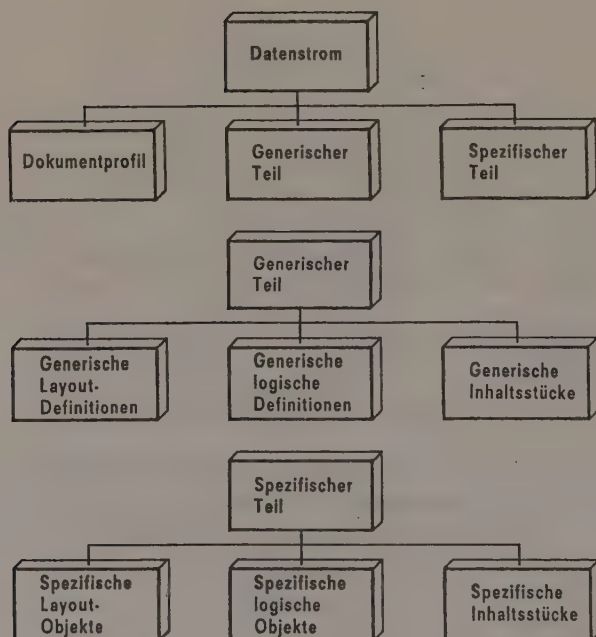


Abb. 2-10: Aufbau des Datenstroms für den Dokumentenaustausch (aus HORAK84b)

Kompatibilitätsstufen. Für die Weiterverarbeitung von Dokumenten nach Übergang in eine andere Arbeitsumgebung sind eine Reihe von *Konformitätsstufen* bzgl. des ODA/ODIF-Standards vereinbart worden (*). Dabei können vornehmlich drei Klassen unterschieden werden (vgl. auch HORAK85b):

- (i) *Wiedergabekonformität:* Zur einfachen Wiedergabe eines Dokuments ist die volle Layoutstruktur, aber auch nur die Beschreibung der Darstellungsseiten hinreichend.
- (ii) *Weiterverarbeitungskonformität:* Hierzu werden die logische Struktur, aber auch die generische logische und Layout-Struktur benötigt. Auf dieser Konformitätsstufe ist die Weiterbearbeitung eines Objekts nach Neuformatierung durch das Zielwerkzeug unter Beachtung der generischen Layout-Struktur möglich. Dieses Ziel kann teilweise durch bloße Angabe der Textstrukturen bis Absatzebene erreicht werden, dabei entspricht die Information den Anforderungen des X.420-Standards.
- (iii) *Format- und Weiterverarbeitungskonformität:* Dabei wird über die Weiterverarbeitungskonformität hinaus eine spezifische Layout-Struktur mitgeliefert, so daß bei der Weiterverarbeitung der neuerliche Formatierungsschritt entfällt und vom ursprünglichen Layout ausgegangen werden kann.

ODA/ODIF vs. DM. Ein Vergleich zwischen ODA/ODIF und DM führt zu folgenden Schlüssen:

-
- (*) Der Begriff der *Konformität* kann in diesem Zusammenhang mit dem der *Kompatibilität* gleichgesetzt werden; allerdings ist ersterer hinsichtlich einer *Norm* und letzterer hinsichtlich eines Werkzeugs zu gebrauchen.

- *Intention:* ODA stellt einen gelungenen ersten Versuch dar, einen detaillierten, universellen Mechanismus für den Aufbau und die Verarbeitung von Dokumenten zu schaffen; der DM-Ansatz hingegen vermeidet Annahmen über Organisationsmodelle von Präsentationsflächen und stellt nur das Grundgerüst für die Einteilung derselben zur Verfügung; damit ist ODA im Verein mit ODIF durchaus geeignet, den Austausch von Dokumenten zwischen Installationen mit völlig divergierenden Software-Umgebungen zu ermöglichen, während der DM-Ansatz aufgrund der Bindung der Typen an Werkzeuge deren Vorhandensein voraussetzt. Man muß allerdings im Falle von ODA ggf. hohe Informationsverluste bei Werkzeugwechsel in Kauf nehmen, da die von manchen Werkzeugen erzeugten Attribute von anderen nicht interpretiert werden können.
- *Organisationsmodelle:* Aufgrund des Anspruchs allumfassender Beschreibung sind die Ausdrucksmöglichkeiten auf der Basis von ODA sehr stark eingeschränkt (auf Text und Rastergrafik, wobei letztere aufgrund der dem Rastermodell inhärenten Geräteabhängigkeit nur bedingt zur Informationsübertragung tauglich ist). Der DM-Ansatz läßt hingegen auf diesem Gebiet weit mehr Freiräume offen.
- *Seitenumbruch:* DM macht keine Aussagen über die Gestaltung des Dokumentinhalts auf den Ausgabenmedien, wohingegen ODA auf dem Begriff der *Seite* als Ausgabeinheit basiert, die dann auch das Typkonzept auf Layout-Ebene entscheidend beeinflusst. Der Seitenumbruch fällt nach dem DM-Ansatz in den Verantwortungsbereich des Werkzeugs, das den Typ des mit dem Dokument identischen Objektes prägt.

- *Typisierung*: Ebenso ist auch die Typisierung beim DM-Ansatz, im Gegensatz zum ausgeprägten Typkonzept in ODA, im Verantwortungsbereich des jeweils zugeordneten Bearbeitungswerkzeuges zu suchen. DM-Typen können im Sinne von Prototyp-Dokumenten definiert werden, wobei die Einhaltung von Layout-Konventionen durch entsprechende Attributierung untergeordneter Rahmen (nicht verschiebbar, nicht teilbar) erreicht werden kann.

Eine abschließende Würdigung sollte keinen der beiden Ansätze in den Schatten stellen, sondern vielmehr die Einsatzgebiete abstecken:

Es ist aussichtslos, zumal im innerbetrieblichen Verkehr oder im Falle enger zwischenbetrieblicher Integration (vgl. MERTENS85a), alle erforderlichen Struktur- und Organisationselemente in generischer Form antizipieren zu wollen. Inner- oder zwischenbetriebliche Absprachen sind erfahrungsgemäß wesentlich flexibler zu handhaben als Normen, die von den relativ trägen Genehmigungsverfahren der zuständigen Gremien abhängen. Dieses Plädoyer soll jedoch nicht den Wert einer ODA-Konformität für einzelne Werkzeuge und Datenstrukturen schmälern, die im allgemeinen öffentlichen Schriftverkehr unabdingbar sind.

3 Die Präsentation

Unabhängig von der Gestaltung der Dokumente selbst ist die Gestaltung der Präsentation der Vorgänge innerhalb eines Büros; der Begriff der *Präsentation* umfaßt dabei nicht nur die visuelle Darstellung von Information, sondern, in Verbindung mit dem Begriff *Interaktion*, auch atomare und zusammengesetzte Handlungen, die die präsentierten Objekte betreffen. Die folgenden Abschnitte enthalten grundsätzliche Betrachtungen zu den angestrebten Zielen auf diesem Gebiet, eine Erörterung der technischen Realisierungsmöglichkeiten und ein Konzept der technischen Realisierung.

3.1 DER DIALOG

Der Begriff *Dialog* bezeichnet den *kontextabhängigen Austausch von Information zwischen zwei Individuen*. Der Bezug auf einen Kontext läßt sich derart deuten, daß durch den Informationsaustausch ein Angleichen des Wissens zu einem Sachverhalt der betroffenen Individuen (*Argumentation und Überzeugung*) oder zumindest ein Aufbau von *Meta-Informationen* (Wissen über das Wissen des Partners; *Kenntnisnahme*) stattfindet. Charak-

teristisch für Dialoge ist also hinsichtlich der Dialogpartner, der ausgetauschten Informationsquanten und des Dialogfortschritts folgendes:

- *Umgebungskontext:* Beide Dialogpartner erwarten voneinander ein Wissen über den Dialoggegenstand (Anwendungsfeld), über Kommunikationsprotokolle (Abbildung der übertragenen Information in Sprache) und das allgemeine Umfeld (Allgemeinwissen). Dabei muß auch über die Inhalte (Intension) und Umfang der einzusetzenden Begriffe Einklang bestehen bzw. erzielt werden.
- *Informationskontext:* Bei der Informationsübertragung wird auf die bereits vorher übertragene Information Bezug genommen; die für eine Aussage erforderliche Kontextinformation muß dabei nicht wiederholt werden.
- *Kontextangleichung:* Das Ziel des Dialogs ist die Angleichung der individuellen Information (in Teilgebieten bis zur Identität).

3.1.1 MENSCH-MASCHINE-INTERAKTIONEN

"Ideally, the personal computer will be designed in such a way that people of all ages and walks of life can mold and channel its power to their own needs."

(KAY77b, S. 230)

"The human user is perceived as a source of trouble, and many of his troubles (if noted) are classified as irrelevant to the functional specification and are dismissed as his own responsibility."

(THIMBLEBY83, S. 72)

Die Interaktion zwischen einem Menschen und einem Rechner ist nicht mit der Interaktion zwischen einem Menschen und einem passiven Medium vergleichbar; durch die Fähigkeit, Information numerisch oder symbolisch zu verarbeiten, kann sich der Rechner durch *simulatives Verhalten* der Arbeitsweise eines fiktiven Dialogpartners annähern.

Wie sieht nun die technische Realisierung von Dialogen aus? DIN84 definiert in Abschnitt 2 der einschlägigen Norm die Grundbegriffe als

- *Dialog*: Dialogschritt oder Folge von Dialogschritten

- *Dialogschritt*: Eingabedaten, zugehörige Verarbeitungsprozesse, zugehörige Ausgabedaten (interessanterweise wird der Kausalzusammenhang zwischen Ein- und Ausgabedaten durch zugehörig charakterisiert).

Als Grundsätze der Dialoggestaltung werden im folgenden angeführt:

- (i) *Aufgabenangemessenheit*: Darunter versteht man den Einbezug der eigentlichen Aufgabe des Benutzers in die Anlage des Dialogs; Belastungen durch Eigenheiten des Systems sollen vermieden werden.
- (ii) *Selbsterklärungsfähigkeit*: Der Dialogpartner, also das System, muß jederzeit Auskunft über Ziel, verfügbare Funktionen und derzeitigen Stand des Dialogs geben können; diese Information steht entweder auf Abruf zur Verfügung (*passive Hilfe*) oder wird zu passendem Zeitpunkt unaufgefordert gegeben (*aktive Hilfe*).
- (iii) *Steuerbarkeit*: Der menschliche Partner hat Einfluß auf Geschwindigkeit, Auswahl und Reihenfolge der einzusetzenden Arbeitsmittel.
- (iv) *Verlässlichkeit*: Das Dialogverhalten des Rechners ist invariant und entspricht den Erwartungen des Benutzers nach Maßgabe des Arbeitsumfelds.
- (v) *Fehlertoleranz und -transparenz*: Fehlerhafte Eingaben werden entweder korrigiert oder zumindest erkannt: Die dabei auszugebenden Fehlermeldungen weisen unmittelbar auf den aufgetretenen Fehler hin.

Der Kommunikationskanal. Über die technische Realisierung, d.h. den Gestaltungsspielraum bei der Umsetzung dieser Anforderungen in technische Realisierungsvorschriften, wird dabei

nichts ausgesagt. FISCHER83 gibt die entsprechenden Hinweise, indem er zunächst das derzeit in der Datenverarbeitung bei herkömmlicher Gestaltung der *Mensch-Computer-Kommunikation* gängige Szenario eines Dialogs beschreibt (vgl. Abb 3-1).

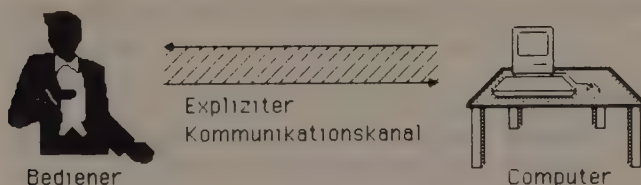


Abb. 3-1: *Herkömmliche Mensch-Computer-Kommunikation*

Im Vergleich mit zwischenmenschlichen Kommunikationsprozessen ist die normalerweise ausschließlich auf visueller Information basierende Mensch-Computer-Kommunikation (wie durch den schmalen Kommunikationskanal gekennzeichnet) als relativ arm anzusehen. Von besonderer Bedeutung ist die Tatsache, daß, von geringem Kontextwissen im Rahmen der Dialogsitzung abgesehen, alle Information *explizit* übermittelt werden muß.

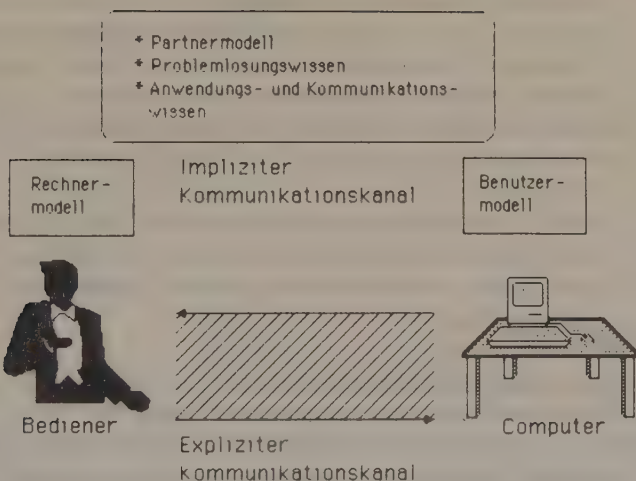


Abb. 3-2: Mensch-Computer-Kommunikation mit implizitem und verbreitertem explizitem Kommunikationskanal

Beiden Mängeln wird im Szenario in Abb. 3-2 abgeholfen: der *explizite Kommunikationskanal* wird durch den Einsatz weiterer Medien auf dem Bildschirm und dem Ansprechen weiterer Sinnesorgane verbreitert. Zudem wird der Rechner durch ein *Benutzermodell* in die Lage versetzt, sich der Denkweise, Ausdrucksweise und dem organisatorischen und sozialen Umfeld des jeweiligen Benutzers anzupassen.

Benutzermodellierung. CARBONELL⁸³ unterscheidet dabei

- *empirische (quantitative) Modelle*, die als Maß für die Kompetenz und das Anpassungsvermögen des Benutzers sein Verhalten (Arbeitsgeschwindigkeit, korrekter Einsatz der verfügbaren Funktionen im Hinblick auf ein Arbeitsziel) ansetzen, d.h. auf Denkvorgänge oder das Wissensreservoir des Benutzers nicht eingehen,
- *analytische (qualitative) Modelle*, die im Gegensatz dazu das kognitive Benutzerverhalten zu simulieren suchen und so auf eine Anpassung des Dialogverhaltens des Systems hinarbeiten.

Die für den Einsatz der ersten Methode erforderlichen Daten sind zwar verhältnismäßig leicht zu ermitteln, lassen jedoch im konkreten Fall diverse Interpretationsmöglichkeiten zu. Der zweite Ansatz hingegen ist in höherem Maße auf den Benutzer bezogen und deshalb noch anfälliger gegenüber den bei der Modellbildung zwangsläufig auftretenden Verallgemeinerungen. Als Kompromiß einer Implementierungsbasis für derartige Systeme schlägt MÖLLER⁸⁶ eine Merkmalkombination vor, die einerseits durch *Metakommunikation* (ein Vorabdialog mit dem Benutzer in spe über soziales und organisatorisches Umfeld sowie Eigenaussagen zu den bisherigen Bedienungserfahrungen) erhoben werden, andererseits auf der *Protokollierung des Benutzerverhaltens* beruhen. Entsprechend dem zugrundegelegten Architekturmodell, das die Funktionen des Rechners auf

- *Ein-/Ausgabeschicht*
- *Dialogschicht und*

- *Anwendungssystemsicht*

verteilt, soll dabei die Protokollierung auf drei Ebenen erfolgen (vgl. hierzu auch HACKER78 (*)):

- *Sensumotorische Ebene*: Auf dieser Ebene werden sensumotorische Fähigkeiten im Sinne automatisierter, hochintegrierter Handlungsmuster, die ohne bewusste Kontrolle ablaufen, beurteilt, die eine besondere Rolle bei der Parametrierung der Ein-/Ausgabeschicht spielen.

Beispiel: Intuitive Benutzung von Positionierungsgeräten (Tasten, Zeigeeinstrumente)

- *Perzeptiv-begriffliche Ebene*: Sie behandelt einfache und gleichbleibende Handlungsmuster, die routinemäßig ausgeführt werden können und die sich in der Dialogschicht durch bei verschiedenen Benutzern verschieden hohe kognitive Repräsentation der formalen Regeln des Dialogablaufs repräsentieren.

Beispiel: Gewohnheitsmäßiges Einsetzen bestimmter Parameter bei der Kommandoeingabe.

- *Intellektuelle Ebene*: Auf dieser Ebene wird die Lösungsfindung für komplexe Probleme betrachtet; insbesondere wird dabei die Einbettung von Einzelhandlungen in einen übergeordneten Zusammenhang und der Übergang von dieser auf die darunterliegenden Ebenen beurteilt.

(*) HACKER78, S. 104f, spricht dabei von *Handlungsregulation* im Sinne der Initiierung bzw. Beeinflussung menschlichen Handelns.

Beispiel: Manuelles Layout einer Dokumentseite.

Wenig bekannt ist derzeit noch über die Umsetzung der so ermittelten *Beschreibungen* in *Beurteilungen*. Das Problem dabei ist, ein gemäß der oben angeführten Klassifikation bestehendes *Kompetenzkontinuum* durch Festlegung von Schwellwerten in ein Klassifikationsschema abzubilden, das aus dem so entstehenden abstrakten Benutzerbild dennoch benutzeradäquate Systemreaktionen erlaubt. Da insbesondere auch das intellektuelle und soziale Umfeld der Benutzer dabei Eingang finden muß, ist man in erster Linie auf Feldversuche größeren Umfanges angewiesen; für eine abschließende Bewertung liegt derzeit noch keine gesicherte Basis vor; weitergehende Betrachtungen zu diesem Thema werden in ROSENOW86 (S. 27ff) präsentiert.

Interaktions- vs. Trägerhandlungen. Untersuchungen zu den elementaren Bürohandlungen führten zu dem Ergebnis, daß Bürotätigkeit auf *Sprechhandlungen* beruht, die durch die Überleitung in entsprechende *Trägerhandlungen* aktualisiert werden können. WEDEKIND85 unterscheidet dabei als Handlungskategorien die

- *Nominativhandlungen (N-Handlungen)*, die als nominative Sprechhandlung den Gegenstand einer Handlung bezeichnen, und die
- *Prädikativhandlungen (P-Handlungen)*, die eine mit dem angesprochenen Handlungsgegenstand auszuführende Tätigkeit angeben.

Handlungen beider Kategorien werden in den moderneren Systemen nicht per sprachlicher Kennzeichnung des anzusprechenden Objektes initiiert, sondern über *Zeigehandlungen* (*deiktische bzw. D-Handlungen*). Dies wird deutlich bei einer Aufzählung der im Bereich der grafischen und textuellen Bearbeitung erforderlichen Interaktionshandlungen. FOLEY84 unterscheidet dabei:

- Selektion von Objekten im Sinne einer N-Handlung,
- Positionierung von Objekten im Sinne einer P-Handlung,
- Ausrichtung von Objekten im Sinne einer P-Handlung,
- Pfadbestimmung als Folge von Positionierungen und Ausrichtungen im Sinne eines komplexen Handlungsschemas,
- Wertbestimmung im Sinne einer N- oder P-Handlung,
- Texteingabe im Sinne einer N- oder P-Handlung.

Aus dieser Liste sind in der Büroarbeit in erster Linie die Bereiche der Selektion, der Werteingabe und der Texteingabe von Bedeutung; die anderen Bereiche werden auf die genannten Gebiete abgebildet.

Bei der Einführung von Trägerhandlungen, die diese Interaktionshandlungen implementieren, ist nun die Abbildung dieser Handlungsarten auf Handlungen mit physischen Geräten der Rechnerperipherie zu bestimmen.

Gestaltungsgrundsätze. WEDEKIND⁸⁵ (S. 16) sieht als obersten Gestaltungsgrundsatz die sog. Engpaßfreiheit des Gesamtsystems an; Engpässe auf der Ebene der Trägerhandlungen müssen dabei dadurch ausgeschlossen sein, daß alle Trägerhandlungen bei hinreichender Begabung des Benutzers auf die sensumotorische Ebene abgewälzt werden können; Engpässe geistiger Art hingegen werden dabei als Verknüpfungsprobleme von N- und P-Handlungen betrachtet, die in Abschnitt 4 noch eine besondere Rolle im Zusammenhang mit dem Begriff der Büroprozedur spielen werden.

Neben dem *Prinzip der unmittelbaren Manipulation* (vgl. hierzu DISESSA86, zu einer Anwendung im Datenbankbereich auch RÖHRLIE85) sowie dem *Prinzip der unmittelbaren Rückkopplung* (vgl. COLEMAN83, S. 23), die beide im Zusammenhang mit dem WYSIWYG-Prinzip (vgl. Abschnitt 2.2.1) von Bedeutung sind, spielt insbesondere das *Prinzip der Modusfreien Arbeitsumgebung* eine entscheidende Rolle. Das Prinzip bezieht sich auf Engpaßsituationen in der perzeptiv-begrifflichen Ebene der Handlungsregulation und "bekämpft" die weitverbreitete *Vorwegnahme* bestimmter Handlungsfolgen durch das Eintreten in einen vom Programmierer vorherbestimmten Zustand mit einer fest vom Programm vorgegebenen Reihe möglicher Nachfolgehandlungen. TESLER81 prägt hierfür den Begriff *Modus (*)*, wobei an anderer Stelle (SMITH82) Teslers Definition wie folgt angegeben wird:

"A mode of an interactive computer system is a state of the user interface that lasts for a period of time, is not associated with any particular object, and has no role other than to place an interpretation on operator input."

Als Lösung für das Dilemma der Vorwegnahme mit gleichzeitiger Einführung von Modi ist das Prinzip "*Selection precedes command*" zu sehen, das in dem hier beschriebenen Umfeld wie folgt gelesen werden kann: "*N-Handlung muß immer vor P-Handlung ausgeführt werden*". In jedem Falle jedoch bleibt die Anwendung dieses Prinzips auf Aktionen innerhalb des Globalkontextes eines Bearbeitungswerkzeugs beschränkt; die Vorwegnahme eines Handlungskontextes durch die Anwahl eines Werkzeugs kann nicht vermieden werden.

Beispiel: Nach der Selektion eines Textstückes in einem

(*) ... wobei der Übergang zu modallogischen Aussagen bei einer formalen Spezifikation des Programmablaufs durchaus erforderlich ist.

Textbearbeitungswerkzeug steht eine Reihe von Bearbeitungskommandos, z.B. löschen, verschieben und Ähnliches zur Auswahl; gleichermaßen ist jedoch die Möglichkeit der Deselektion und Neuselektion eines anderen Objektkomplexes mit ggf. anderen anzuwendenden Operationen nicht vorweggenommen. Im umgekehrten Fall, der Auswahl einer Operation mit anschließender Parametrierung durch Objektkennzeichnungen, ist die Menge der möglichen auswählbaren Objekte bereits derartig eingeschränkt, daß hier ein Engpaß auftreten kann.

3.1.2 ZUM DIALOGBEGRIFF

"The problem of the language speaker is to get the directed attention of an unprepared hearer and slide some information into his mind in a very short time."

(MCDERMOTT76, S. 6)

Der Begriff *Dialog* wurde in den bisherigen Betrachtungen, namentlich in Anbetracht der Behandlung von Dialogelementen des Mensch-Rechner-Dialogs, als Informationsaustausch zwischen Subjekten mit folgenden Randbedingungen verwendet:

- Abhängigkeit von einem gemeinsamen Kontext,

- die Berücksichtigung des Wissens um den Partner und dessen Umfeld,
- das Befolgen von Konventionen,
- das Streben nach einem gemeinsamen Ziel.

Ziel dieses Abschnittes soll sein, Ansätze zur Rekonstruktion des Begriffs im allgemeinen Sprachgebrauch zu liefern und damit den speziellen Umstand im Mensch-Rechner-Dialog, nämlich die Heterogenität der Dialogpartner, angemessen zu würdigen; zugleich soll über ein Klassifikationsschema für abstraktere Dialogelemente Einsicht in die Anforderungen an entsprechende Softwarekomponenten gewonnen werden.

Der Dialog als Spielsituation. Theoretische Betrachtungen des Phänomens *Dialog* suggerieren eine Betrachtungsweise, die sich an spieltheoretische Betrachtungen anlehnt (vgl. hierzu LORENZEN⁷⁸, S. 31ff). Die dabei betrachtete Spielklasse fällt unter die Kategorie der *Offenen Zweipersonenmattspiele*, die folgendermaßen definiert sind:

Gegeben sei

- eine Menge der *Spielstellungen*,
- eine auf den Spielstellungen deklarierte Relation *T*, die die möglichen Übergänge von einer Spielstellung in eine Folgestellung beschreibt und somit als *Spielregel* betrachtet werden kann,
- zwei *Spieler* zur Auswahl des jeweils durchzuführenden Zuges gemäß der Spielregel,

- eine *Klasseneinteilung* der Spielstellungen dahingehend, daß die Spielstellungen disjunkten Zugbereichen zugeordnet werden und neben der daraus resultierenden Rollenzuordnung an die Spieler auch ein abwechselnde Zugfolge festgelegt wird,
- eine Menge von *Anfangsstellungen*,
- eine Menge von *Endstellungen*.

Bezogen auf die Anwendung bei der Rekonstruktion des Dialogbegriffs heißt dies:

- Die Rollenverteilung innerhalb des Dialogspiels sieht einen der Spieler als den *Proponenten* einer Aussage, der eine Aussage ggf. durch Angabe von Regeln und Behauptung ihrer Prämissen zu belegen sucht (*Behaupten*), den anderen als *Opponenten*, der im Zuge eines Validierungsprozesses die Gültigkeit der Aussage hinterfragt (*Bezweifeln*).
- *Anfangsstellungen* sind durch eine vom Proponenten postulierte Aussage charakterisiert, *Endstellungen* durch die Bestätigung des vom Proponenten im Zuge des Dialogs aufgestellten Aussagesystems durch den Opponenten.
- Gültige Spielstellungen sind auf Seiten des Proponenten durch eine Menge von mit Hilfe von Prädikaten, Junktoren (Konjunktion, Disjunktion, Negation, Implikation) erstellten Aussagen charakterisiert; seitens des Opponenten ist entsprechend die Negation von Teilaussagen bzw. Implikationen zulässig. Für den ersten Fall führt GETHMANN79b den Terminus *Horizontale Begründungsdimension*, für den zweiten Fall *Vertikale Begründungsdimension* ein. Entsprechend sind für den

Dialogverlauf (*) eine *reduktive* Begründungsrichtung bei rückschreitender Auflösung eines Postulats und eine *produktive* Begründungsrichtung bei fortschreitendem Einsatz der erzielten Ergebnisse als Prämissen für neue Postulate; Abb. 3-3 zeigt die Grundformen dieser Begründungsdimensionen. Entsprechende Techniken sind im Bereich der Künstlichen Intelligenz unter den Termini *Backward Chaining* und *Forward Chaining* bekanntgeworden (vgl. u.a. NILSSON82, S. 34 oder WINSTON84, S. 152).

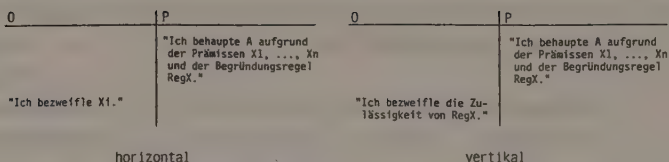


Abb. 3-3: Diskursformen in der vertikalen und horizontalen Begründungsdimension

 (*) GETHMANN79b gebraucht in diesem Zusammenhang den Terminus *Diskurs* anstelle von *Dialog*.

In GETHMANN79b wird eine umfassende Kategorisierung von typischen Dialogsituationen mit einer weiteren Unterscheidung nach Anfangssituation (Eröffnung), Verlauf und Endsituation (Abschluß) des Dialogs angegeben; Abb. 3-4 zeigt eine erweiterte Darstellung, die im folgenden in die typischen Dialogformen des Mensch-Computer-Dialogs übergeführt wird. Dabei symbolisieren die Partikel BEH die Sprechhandlung des Behauptens und BEZW die Sprechhandlung des Bezweifeln.

		Eröffnung		Verlauf		Abschluß	
horizontal	reduktiv	O ... BEZW(A)	P BEH(A) BEH(A,R(A,B))	O ... BEZW(B)	P ... BEH(B,R(B,C))	O ... BEH(C)	P BEH(C,R(C,D)) ...
	produktiv	O BEH(C) ... BEZW(A) ...	P BEH(C) BEH(A) BEH(A,R(A,B))	O BEZW(B) BEZW(C)	P ... BEH(B,R(B,C))	O BEH(C) ... BEZW(B) BEH(B)	P BEH(C) ... BEH(B,R(B,C))
vertikal	reduktiv	O ... BEZW(A)	P BEH(A) BEH(A,R(A,B))	O BEZW(R)	P BEH(R,S(R,T))	O ... BEH(R)	P BEH(R,S(R,T))
	produktiv	O BEH(C) ... BEZW(A) ...	P BEH(C) BEH(R) BEH(A,R(A,B))	O BEZW(R) BEZW(S)	P ... BEH(R,S(R,T))	O BEH(C) ... BEZW(R) BEH(R)	P BEH(C) ... BEH(R(C))

Abb. 3-4: Klassifikationsschema für Standarddiskurse

Wie stellt sich nun die Rollenverteilung nach diesem Schema zwischen Benutzer (Mensch) und Computer dar? Die Prämisse für derartige Betrachtungen ist zunächst, daß jeglicher Informationsaustausch zwischen Mensch und Rechner begründungs- und da-

mit diskursbedürftig ist. Abb. 3-5 zeigt die in Abb. 3-4 skizzierten Standardsituationen in ihrer Beziehung zu typischen Dialogformen des Mensch-Rechner-Dialogs.

	Eröffnung	Verlauf	Abschluß
horizontal	reduktiv Auftreten eines Fehlers	Systemseitige Begründung von Fehlerzuständen	Zufriedenstellung des Benutzers
	produktiv Eingabe von Daten	Begründung/Bestätigung bei der Plausibilitätskontrolle	Obernahme der Daten
vertikal	reduktiv Aufstellen einer Hypothese in einem wissensbasierten Dialogsystem	Regelbasierter Dialog	Bestätigung/Ablehnung der Hypothese
	produktiv Entwurf einer Datenstruktur	Darlegung der verwendeten Konstruktoren	Obernahme/Ablehnung der Datenstruktur

Abb. 3-5: Typische Situationen im Mensch-Computer-Dialog

Man sieht, daß sich der Mensch überwiegend in der Rolle des Proponenten befindet: Die Aufnahme eingegebener Daten sowie die Korrektur von Fehlern unterliegt der Kontrolle des Systems. In den Fällen, in denen der Mensch als Opponent auftritt, sind in erster Linie technisch bedingte Ausnahmesituationen als Ursache zu sehen.

Dialogformen in der Mensch-Computer-Kommunikation. Geht man nun von den betrachteten Dialogformen auf Anwendungen im Bereich der Mensch-Computer-Kommunikation über, lassen sich zunächst vier prinzipielle Dialogkategorien im Sinne einer funktionalen Klassifikation unterscheiden (vgl. hierzu auch VIERECK86):

- der *Anwendungsdialog*, der der Verwirklichung des eigentlichen Ziels bei der Benutzung des Anwendungsprogramms dient,
- der *Kontrolldialog*, der die verwaltungstechnischen Teile der Dialogeröffnung, -beendigung sowie des Auf- und Abbaus der Arbeitsumgebung regelt,
- der *Gestaltungsdialog*, der die Präsentation der Information auf dem Ausgabemedium regelt, und
- der *Metadialog*, der die Funktionsweise des Dialogsystems selbst erläutert und ggf. dem Benutzerverhalten anpaßt.

Die Formulierungsebenen für Dialoge umfassen:

- die *Abstrakte Dialogstruktur*, die eine Abfolge einzelner *Abstrakter Dialogformen* (*Datenabfrage*, *Auswahlangebot*, *Kommando*, *Unterbrechung*) darstellt,
- die *Konkrete Dialogstruktur*, die beschreibt, wie Interaktionen darstellungstechnisch ablaufen, und
- die *Dialogrealisierung*, die die Zuordnung physischer Ein-/Ausgabegeräte zu den Interaktionen umfaßt.

Der folgende Abschnitt erläutert ein umfassendes Gesamtkonzept, das die Realisierung eines Systems zur Einbettung von Anwendungsdialogen beschreibt.

3.2 DIE SCHREIBTISCHOBERFLÄCHE

Gravierenden Einfluß auf die Gestaltung von Benutzerschnittstellen im Bürobereich hat die Metapher der *Schreibtischoberfläche als Bearbeitungsmedium*. Die Grundidee dabei ist die Vorstellung, alle Bürovorgänge in einem Büro auf eine Schreibtischoberfläche abzubilden, d.h. die Abläufe und Ablagestrategien aus einem dreidimensionalen Koordinatensystem in ein zweieinhalbdimensionales Koordinatensystem (zweidimensional mit Überlagerungspräzedenz) abzubilden (vgl. hierzu SMITH82 oder ELLIS82). Treibende Kraft hierbei war die Vorstellung, die gewohnte Benutzerumgebung möglichst naturgetreu zu simulieren und insbesondere die Akzeptanzschwelle bei Einführung derartiger Systeme dadurch zu senken, daß alle für Entscheidungsträger auf den verschiedenen Ebenen der Informationspyramide relevanten und von der herkömmlichen Arbeitsumgebung vertrauten Arbeitsstile weiterhin unterstützt werden konnten. Die erwähnten, besonders zu berücksichtigenden Arbeitsstile sind in den meisten Fällen komparativer bzw. korrelativer Natur (für eine vertiefende Behandlung siehe u.a. DÖRNER79, S. 13 und S. 103ff). Implementierungen dieses Prinzips gehen in den meisten Fällen auf das wegweisende System Xerox 8010 zurück, die Grundzüge wurden jedoch im Rahmen der Entwicklung des Dynabook, eines Experimentalrechners der Learning Research Group in Palo Alto CA konzipiert (vgl. KAY77a).

Im folgenden werden die Anforderungen an ein derartiges System skizziert und gleichzeitig ein terminologisches Grundgerüst für den Aufbau entsprechender Systeme gelegt; u.a. bei der Lektüre der Beiträge in HOPGOOD86 zeigt sich, daß die Terminologie auf diesem Gebiet derzeit noch sehr unsauber und widersprüchlich aufgebaut ist. Weitere Abschnitte dieses Kapitels sind der Umsetzung in ein Implementierungskonzept gewidmet.

3.2.1 DAS UMFELD

Untersuchungen der Arbeit im Büro (MALONE83, SUCHMAN83) zeigen, daß die Organisation der Information innerhalb des Büros individuellen Kriterien hinsichtlich Ordnung, Übersicht und Aufgabenangemessenheit unterliegt. Besonders die räumliche Anordnung muß dabei dem Benutzer des Arbeitsplatzes überlassen bleiben, um so mehr, wenn sein Aufgabengebiet eher der unstrukturierten Aufgabenkategorie zuzuordnen ist. Demnach muß das Design dem Prinzip der *Simulation der gewohnten Benutzerumgebung* folgen, wobei dem simulativen Akt ein rekonstruktiver im Sinne einer Rückführung der angetroffenen Phänomene auf einige wenige Basiskonzepte vorangehen muß. Zur Verdeutlichung zeigt Abb. 3-6 beispielhaft die Aufteilung der simulierten Schreibtischoberfläche am Arbeitsplatz eines Software-Entwicklers.

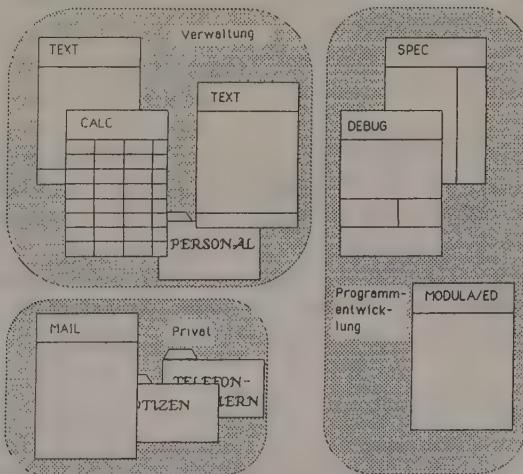


Abb. 3-6: Beispiel für die Schreibtischoberfläche eines Programmentwicklers

Es ist noch nicht abschließend geklärt, ob das menschliche Aufnahmevermögen und Gedächtnis eher für ein Auffinden von bestimmten Objekten anhand des *Namens* oder eher für die Rückbesinnung auf die *räumliche Anordnung* geeignet ist. Untersuchungen zu Ablageverfahren (dabei insbesondere JONES86) scheinen ersteres zu indizieren. Dennoch ist die *räumliche Anordnung* zumindest für den Aufbau individueller Beziehungen zwischen Objekten (z.B. die Zonenbildung in Abb. 3-6) unabdingbar und führt im Extremfall zur Bildung von Stapeln, wie sie in MALONE83 dargestellt ist.

Die Präsentation. Grundelemente der Schreibtischoberfläche sind:

- *aktive Objekte* in Form von Präsentationsflächen, die von einem Applikationsprogramm bzw. einem Bearbeitungswerkzeug beschickt werden,
- *passive Objekte*, die die interne Repräsentation eines Bearbeitungszustandes enthalten und auf der simulierten Schreibtischoberfläche als sog. *Piktogramme* (*) dargestellt werden,
- *Strukturierungsobjekte*, die der visuellen Darstellung einer Kataloghierarchie dienen; dabei wird in den meisten Systemen das Bild des *Aktenordners* zur Verdeutlichung eingesetzt, in einigen Fällen als rekursives Strukturelement, in anderen ergänzt um übergeordnete Strukturierungselemente wie *Regal*, *Aktenschrank* auf den höheren Ebenen der Strukturierungshierarchie.

(*) STAUFER85 gibt u.a. wertvolle Hinweise zur Gestaltung von Piktogrammen in Form von gestaltpsychologischen Kriterien (S. 50-60) sowie eine Methodologie zu deren Design (S.135ff).

Besondere Beachtung verdient die Behandlung der aktiven Objekte, da hier sowohl konzeptionell wie implementierungstechnisch der wohl größte Aufwand zu treiben ist.

Aktive Objekte entsprechen Applikationsprogrammen, die zum Betrachtungszeitpunkt in der Bearbeitung eines Dialogschrittes begriffen sind oder diesen abgeschlossen haben und auf eine Neueingabe warten. Die Darstellung auf der Bildschirmoberfläche spiegelt deshalb den derzeitigen Programmzustand bzw. den Zustand der zugrundeliegenden Datenstruktur wider. Da, wie bereits in den grundsätzlichen Betrachtungen zur Büroumgebung in Abschnitt 1.4 angedeutet, die Möglichkeit mehrerer parallel aktiver Arbeitsumgebungen gewahrt werden muß, werden derartige Umgebungen über einen Filtermechanismus, visuell sichtbar als Fenster, präsentiert (in ihrem derzeitigen Begriffsinhalt eingeführt in KAY77a; eine vergleichende Würdigung derzeit marktverfügbarer Fenstersysteme bietet BALZERT85). Da innerhalb eines Anwendungskontextes häufig mehrere, zwar lose zusammenhängende, jedoch technisch unabhängig zu realisierende Aspekte eines Bearbeitungszustandes sichtbar werden müssen, liegt die Erweiterung des Fensterbegriffs zu dem des *Partitionierten Fensters* (*paned window*) nahe. Abb. 3-7 zeigt ein Beispiel für die Partitionierung anhand einer Momentaufnahme eines Verwaltungssystems für Elektronische Post: eine Partition zeigt Kataloginformation, während die andere den Detailblick in eine Nachricht ermöglicht (*).

(*) Diese Technik ist im Umfeld von Arbeitsplatzrechnern der höchsten Leistungsklasse als *Browsing* bzw. *Cascaded Filtering* populär geworden (vgl. dazu u.a. GOLDBERG79, S. 151 oder GOLDBERG83, S. 297ff).

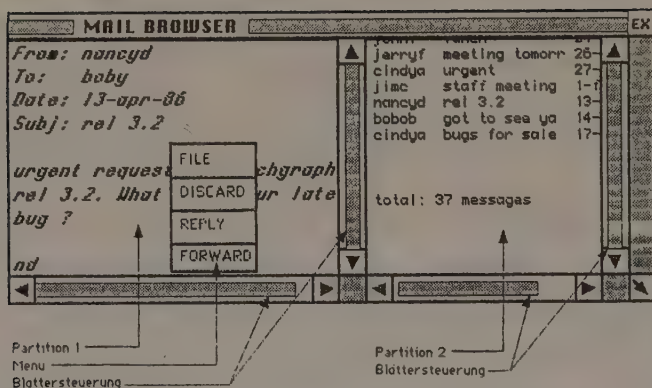


Abb. 3-7: Beispiel für ein Partitioniertes Fenster

3.2.2 ALLGEMEINE ANFORDERUNGEN

Die Anforderungen an ein Verwaltungssystem zur Simulation einer Schreibtischoberfläche beruhen auf Prämissen, die die Anwendbarkeit dieses Systems für Anwendungen fordern, die sowohl unter der Annahme eines entsprechenden zugrundeliegenden Systems erstellt wurden, als auch für Anwendungsprogramme, die ein

beliebiges logisches oder physisches Ausgabegerät voraussetzen. Das beschriebene System wird dabei im folgenden kurz *GDM* (*Generalized Desktop Manager*) genannt; eine direkte Integration der verwendeten Werkzeuge untereinander ist explizit nicht gefordert. Das System muß sowohl die Funktionen herkömmlicher Fensterverwaltungssysteme wie auch Objekt- und Applikationsprogrammverwaltung abdecken. Dabei muß das zu entwickelnde Instrumentarium die Anforderungen einer transparenten Benutzeroberfläche sowie der Büroumgebung weitestgehend berücksichtigen.

Zuordnung der Objekte zu Manipulationswerkzeugen. Auf der Schreibtischoberfläche ist der Begriff des Manipulationswerkzeuges bzw. Anwendungsprogramms nicht bekannt. Manipulationswerkzeuge werden nur als Implementierungen des Funktionsvorrates eines Objekttyps angesehen, die das Verhalten und die Eigenschaften der Instanzen dieser Klasse, die allgemein der Kategorie der Objekte angehören, festlegen. Systemseitig ist die Möglichkeit der Neuschaffung von Objekten gewährleistet, wobei man *transiente* Objekte und *permanente* Objekte unterscheiden muß, wobei erstere nur als aktive, letztere auch als passive Objekte existieren können und einer Repräsentation des letzten Bearbeitungszustands bis zur Re-Aktivierung bedürfen.

Kompatibilität. Wie bereits erwähnt, können Anwendungsprogramme in unterschiedlichem Grad die Dienste von GDM ausnutzen; im Falle der Portierung einer datenendgerätbezogenen Anwendung werden die Ein-/Ausgabeoperationen auf entsprechende Dienste einer Laufzeitbibliothek abgebildet.

Entkopplung des Anwendungsprogramms von der Organisation der Schreibtischoberfläche. Der Benutzer und Verwalter der Schreibtischoberfläche ist frei in der Anordnung der aktiven und inaktiven Objekte auf der Schreibtischoberfläche. Indes, keine dieser Manipulationen hat Auswirkungen auf die Anwendungsprogramme und muß von diesen berücksichtigt werden. Desgleichen ist

ein Blättern im Inhalt einer Fensterpartition nicht notwendigerweise mit einer Informationsanforderung beim Anwendungsprogramm verbunden. Das einzige Bindeglied zwischen Applikationsprogramm und Außenwelt ist eine Präsentationsfläche mit zugrundeliegender Datenstruktur und einem Exekutionsmodell, das die visuelle Darstellung dieser Struktur regelt. Alle Informationen bezogen auf Benutzereingriffe in die Schreibtischorganisation oder hinsichtlich der Sichtbarkeit bestimmter Informationsquanten sind verfügbar; es steht dem Anwendungsprogramm frei, dies auszunutzen oder nicht.

Überlappungs- vs. Kacheltechnik. Eine der fundamentalen Diskussionen im Bereich der Präsentationstechnik durch Fenster ist zwischen den Anhängern der sog. *Überlappungstechnik* (*window overlap*) und den Jüngern der sog. *Kacheltechnik* (*tiling*) im Gange. Argumente zu den jeweiligen Standpunkten finden sich u.a. in MORRIS86, COHEN86, GOODFELLOW86 und STRECK85. In summa kann der Stand der Diskussion derzeit folgendermaßen beurteilt werden:

- o Die *Kacheltechnik* nimmt eine *zweidimensionale* Aufteilung des Bildschirms vor, d.h., auf ein Überlappen der Darstellungsbereiche wird explizit verzichtet und gleichzeitig der physisch vorhandene Raum voll genutzt. Die Neueröffnung bzw. Freigabe eines Fensterbereiches zieht Größenveränderungen der benachbarten Fenster nach sich. Vorteile dieser Technik sind:
- Die Programmierung dieser Technik ist einfach und effizient, da keine komplexen Clipoperationen durchzuführen sind und Ein-/Ausgabeoperationen einfach den entsprechenden Prozessen zugeordnet werden können,

- Nur in seltenen Fällen muß das Bild neu aufgebaut werden.
 - Die Technik verlangt nur selten und kaum detailliert Eingriffe des Benutzers zur Gesamtaufteilung.
 - Die gesamte verfügbare Information ist zu jedem Zeitpunkt sichtbar.
- o Die Überlappungstechnik ermöglicht eine zweieinhalb-dimensionale Einteilung des Bildschirms, die neben der Aufteilung im zweidimensionalen Raum noch eine Präzedenzstruktur der Fenster bei der Abbildung auf der Bildschirmoberfläche berücksichtigt. Neueröffnung und Freigabe von Fenstern beeinträchtigen andere Fenster in keinem Fall durch erzwungene Größenveränderungen, behindern aber ggf. die Sichtbarkeit bestimmter Informationen durch Überlagerung. Hieraus folgt, daß ein gewisser Aufwand für eine arbeitsgerechte Organisation der simulierten Schreibtischoberfläche getrieben werden muß. Vorteile dieser Technik sind:
- Der Benutzer kann die Fensteranordnung frei wählen.
 - Die Anordnung ist kaum durch technische Restriktionen des Systems eingeschränkt.
 - Die Anordnung und darstellbare Informationsmenge sind von physischen Eigenschaften des Bildschirms entkoppelt.

GDM bedient sich einer Hybridtechnik: Die Partitionen innerhalb der Fenster werden nach der Kacheltechnik verwaltet, da an-

genommen wird, daß diese Partitionen bei Ein-/Ausgabeaktivität über dieses Fenster einem gemeinsamen Zweck dienen und deshalb sichtbar sein müssen. Die zugrundeliegende Fixierung eines 1:1-Verhältnisses der Fenster zu Applikationsprogrammen unterstützt diese Annahme. Die Schreibtischoberfläche ordnet die Fenster gemäß der Überlappungstechnik an, um so einerseits die Illusion einer den herkömmlichen Verhältnissen angepaßten Arbeitstechnik nicht zu gefährden und um andererseits eine stapelorientierte Arbeitstechnik zu unterstützen. Dabei wird der gravierendste Nachteil der Kacheltechnik in dieser Anwendung unbedeutend: Durch die Bezugnahme auf ein Fenster variabler Größe ist eine Abhängigkeit der Kachelgröße von der übergeordneten Einheit (im schlimmsten Fall, wie z.B. in MORRIS86, vom Bildschirm) nicht mehr gegeben, da diese Bezugseinheit ja wiederum innerhalb der übergeordneten Einheit variabel angelegt ist und aufgrund der dort angewendeten Überlappungstechnik, wie bereits erwähnt, keine Interdependenzen zwischen verschiedenen Fenstern hinsichtlich der Größe auftreten können.

Zentrale Katalogverwaltung. GDM verwaltete zentral die Zuordnungen von Bearbeitungswerkzeugen zu Objekttypen, die entsprechenden Fensterkonfigurationen und die räumliche Anordnung von inaktiven Objekten auf der Schreibtischoberfläche. Die dafür vorgesehene zentrale Datenstruktur heißt GDM-Katalog.

Unterstützung der Benutzerschnittstelle. Ein integraler Bestandteil des GDM-Systems ist die kontextsensitive, menügesteuerte Dialogunterstützung.

Portierbarkeit des Systems und der Bearbeitungswerkzeuge. Die Architektur von GDM beruht auf folgenden Voraussetzungen hinsichtlich der zugrundeliegenden Konfiguration:

- Arbeitsplatzrechner mit hoher CPU-Leistung, virtueller Speicherverwaltung und Positionierungsgeräten wie z.B. Maus, Tablett etc.,
- Lokale Kommunikationsdienste zwischen den Rechnern mit transparentem Zugriff auf zentrale Datenbankdienste (DB-Server),
- Hochauflösender Rasterbildschirm,
- effiziente Interprozeß-Kommunikationsdienste.

Die Übertragbarkeit von GDM ist deshalb in erster Linie von den Charakteristiken des Bildschirms (Auflösung) abhängig, gleichermaßen jedoch von der Funktionalität des zugehörigen Bildschirmprozessors (Speicherfähigkeit, Mächtigkeit der Rasteroperationen). Soweit Anwendungsprogramme nicht auf Rasterebene operieren, sondern Objekte auf höherem Abstraktionsniveau manipulieren, ist eine verhältnismäßig einfache Portierbarkeit gewährleistet.

3.2.3 DIE ARCHITEKTUR

Die Abbildung von der Ein-/Ausgabeschnittstelle auf den Bildschirm des Arbeitsplatzrechners ist in fünf Schichten gegliedert. Diese Schichten korrespondieren zu fünf Einheiten des Abbildungsprozesses, die im folgenden Präsentationsfläche (*pad*), Partition (*pane*), Fenster (*window*), Virtuelle Schreibtischoberfläche (*virtual desktop*) und Physische Schreibtischoberfläche (*physical desktop*) genannt werden. Abb. 3-8 zeigt schematisch den Abbildungsprozeß von den Darstellungsflächen bis zu dem auf dem physischen Gerät abgebildeten Ausschnitt der simulierten Schreibtischoberfläche.

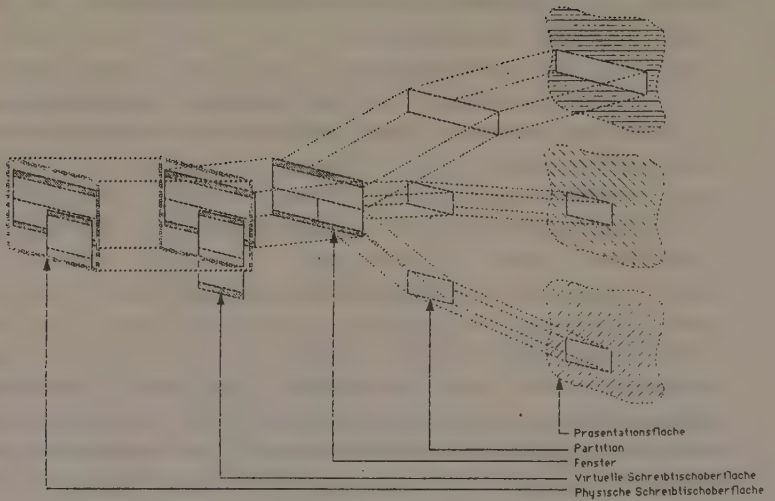


Abb. 3-8: Der Abbildungsprozeß

Die Funktionen jeder dieser Schichten werden auf Verwaltungsmodule abgebildet, die bei n aktiven Fenstern bzw. Anwendungsprogrammen im Rahmen von $n+1$ Prozessen instantiiert werden. Dabei werden die drei anwendungsnahen Schichten als *Lokales GDM-Verwaltungsprogramm* (*local GDM manager*) integriert in das Laufzeitsystem des Anwendungsprogramms abgelegt, die zwei gerätenahen Schichten als *Globales GDM-Verwaltungsprogramm* (*global GDM manager*) in einem zentralen Prozeß dem Ausgabegerät beigegeben. Eine Übersicht über die Architektur gibt Abb. 3-9.

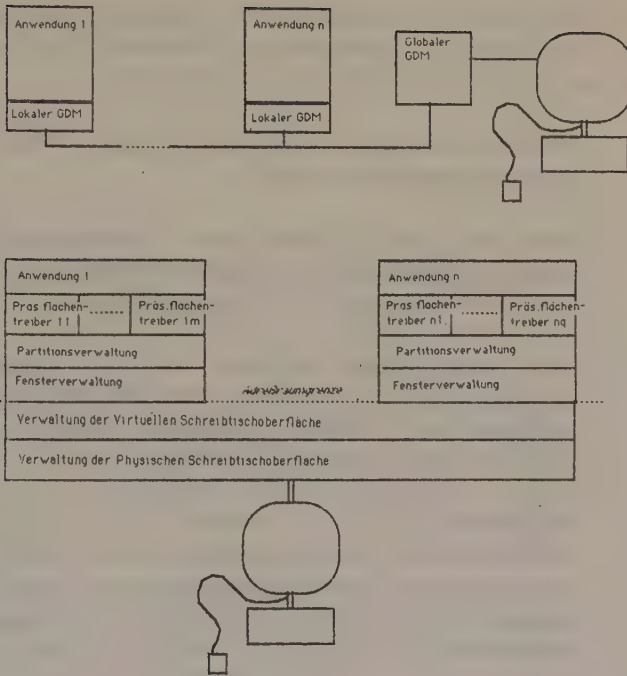


Abb. 3-9: Modularisierung in GDM

Die Präsentationsfläche (pad). Präsentationsflächen bilden die Schnittstelle zwischen den Bearbeitungswerkzeugen und GDM. Für den Benutzer entsprechen sie Burodokumenten, die durch das jeweils aktive Bearbeitungswerkzeug verwaltet werden. Anwendungsprogrammseitig sind sie eine beliebig große, endliche Darstellungsfläche, die im Rahmen eines Organisationsmodells verwaltet wird und die entsprechenden Objekte bereitstellt (vgl. dazu auch Abschnitt 2.3). Wichtig ist auch hier wiederum das

ausgabespezifische Koordinatensystem zur Platzierung und Lokalisierung von Objekten sowie das Darstellungsmodell für diese Objekte. Die geleisteten Dienste sind im einzelnen:

- Bereitstellung einer Laufzeitbibliothek gemäß dem gewählten Organisationsmodell,
- Verwaltung einer Datenstruktur entsprechend der Abstraktionsebene des Organisationsmodells, die den Inhalt der Präsentationsfläche widerspiegelt,
- Clip- und Konvertierungsdienste zur Erstellung einer rasterisierten und den zu einem gegebenen Zeitpunkt gültigen Partitionsgrößen angepaßten Darstellung in einem geräteabhängigen Koordinatensystem (*),
- Konvertierung der Eingabedaten, z.B. Zeichen, Funktionstasten, Lokalisierungskoordinaten gemäß der Interpretation des Organisationsmodells (z.B. Koordinatenangaben in Zeichen und Spalten statt absoluter Bemaßung).

Je nach Organisationsmodell obliegt die Realisierung einem Treiberprogramm (*pad driver*); Abbildung 3-10 zeigt den Datenfluß durch diese Schicht.

- (*) Anfordernde Schichten werden derart in die Lage versetzt, ggf. inkrementell Teile der Präsentationsfläche in rasterisierter Form anzufordern; aufgrund der angewendeten Überlappungstechnik sind oft nur Teile der Präsentationsflächen sichtbar und unterliegen somit Veränderungen.

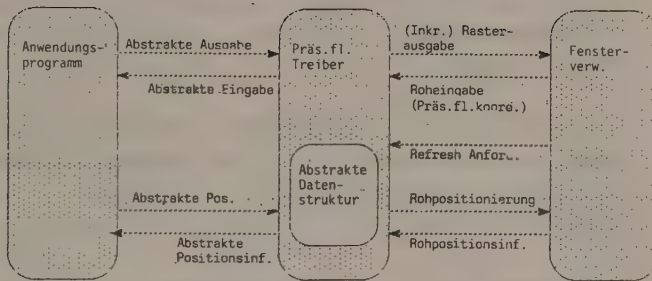


Abb. 3-10: Datenfluß durch die Präsentationsflächen-Schicht

Die Partition (pane). Wie bereits angedeutet, gibt eine Partition als Teil eines Fensters einen Ausschnitt der zu einem gegebenen Zeitpunkt gültigen Präsentationsfläche wieder. GDM steuert zudem Positionierungshilfen zur Einstellung des gewünschten Ausschnitts bei (scroll bars in Abb. 3-7). Die von dieser Schicht angebotenen Dienste umfassen folgende Funktionen:

- horizontales und vertikales Blättern, sowohl auf Eingriff des Benutzer als auch auf Anforderung des Anwendungsprogrammes hin (in den meisten Fällen zum Abgleich der Inhalte der verschiedenen Partitionen eines Fensters),
- Information der Präsentationsflächen-Schicht über den derzeitigen Status der sichtbaren Information, entweder durch Funktionsaufruf oder durch Trap-Mechanismen bei Veränderungen,
- Konvertierung von Eingabekoordinaten aus einem partitionsspezifischen in ein präsentationsflächenspezifisches Koordinatensystem (bei Eingaben über Zeigeeinstrumente),
- Behandlung der auf Partitionsebene definierten Kommandomenüs.

Abb. 3-11 stellt den Datenfluß durch die Partitionsschicht dar.

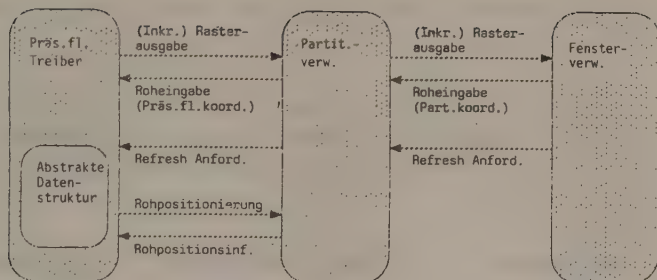


Abb. 3-11: Datenfluß durch die Partitionsschicht

Das Fenster (window). Fenster sind Komplexobjekte, die durch einzelne Partitionen aufgebaut sind; nach den Entwurfsprinzipien von GDM sind sie die Träger der Externen Repräsentation (bildschirmseitig) eines Büroobjektes. Die in einem Fenster zusammengefaßten Partitionen werden nach der Kacheltechnik organisiert, d.h., Modifikationen des Layouts können nur durch Verschieben der Partitionsgrößen bei gleichzeitiger Größenänderung der Nachbarpartitionen erfolgen. Zugleich kann aber auch die Größe des Fensters insgesamt geändert werden und dadurch die an den Rändern platzierten Partitionen in der Größe modifizieren.

Im einzelnen erbringt die Fensterschicht folgende Dienste:

- Verwaltung des Layouts der Partitionen mit der Veranlassung der entsprechenden Restaurationsanforderungen für Partitionsinhalte,
- Konvertierung fensterbezogener Koordinaten auf partitionsbezogene Koordinaten (je nach Layout),
- optionale Abspeicherung eines rasterisierten Abbilds des Fensters zur Reduzierung von Restaurationsanforderungen von Seiten der Schreibtischoberflächenverwaltung,
- Behandlung der auf Fensterebene definierten Kommandomenüs.

Abb. 3-12 zeigt eine Übersicht über den Datenfluß durch die Fensterschicht.

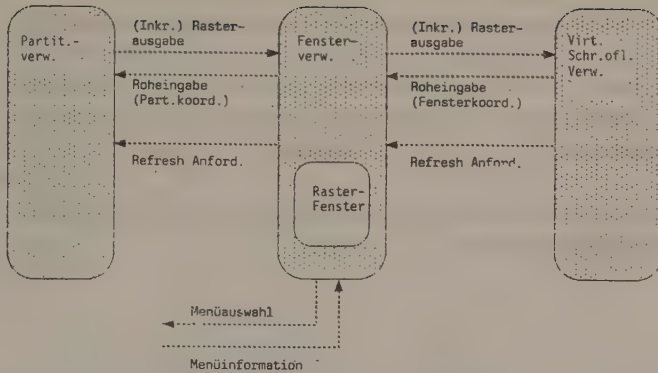


Abb. 3-12: Datenfluß durch die Fensterschicht

Die Virtuelle Schreibtischoberfläche (virtual desktop). Waren bisher nur Aspekte der Fensterverwaltung von Bedeutung, wird auf der Ebene der Virtuellen Schreibtischoberfläche das gesamte Spektrum der Simulation der Benutzerumgebung erfaßt. Dabei werden die aktiven und inaktiven Objekte auf einer beliebig großen, aber endlichen Fläche verwaltet und gemäß den Regeln der Überlappungstechnik ggf. überlagert. Die Dienste dieser Schicht umfassen folgende Funktionen:

- Platzierung, Verschiebung und Neuanlage aller Büroobjekte
- Überwachung des konsistenten Bildaufbaus bei Überlappungen
- Hilfen zum schnellen Auffinden von Objekten (Katalog)
- Konvertierung von Koordinaten bezüglich der Virtuellen Schreibtischoberfläche auf Fensterkoordinaten im Eingabefall.

Abb. 3-13 stellt den Datenfluß durch die Schicht der Virtuellen Schreibtischoberfläche dar.

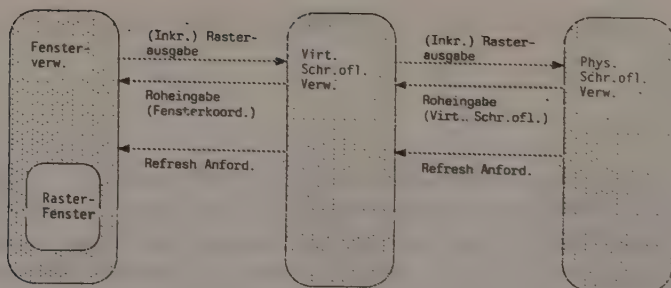


Abb. 3-13: Datenfluß durch die Schicht der
Virtuellen Schreibtischoberfläche

Die *Physische Schreibtischoberfläche* (*physical desktop*). Die Physische Schreibtischoberfläche ist ein Ausschnitt der Virtuellen Schreibtischoberfläche gemäß der Größe des tatsächlich vorhandenen Geräts. Ein-/Ausgabeoperationen werden nach Gerätekoordinaten berechnet und entsprechend den gültigen Transformations- und Streckungsparametern auf Koordinaten der Virtuellen Schreibtischoberfläche umgerechnet. Zur Veränderung des Betrachterstandpunktes stellt sie Navigationshilfen (*panning*) zur Verfügung.

Abb. 3-14 bietet eine Übersicht über den Datenfluß durch diese Ebene.

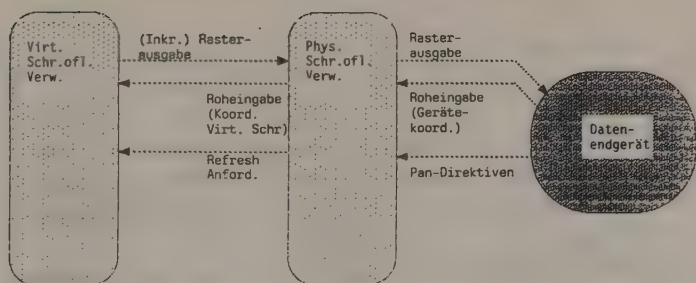


Abb. 3-14: Datenfluß durch die Schicht der Physischen Schreibtischoberfläche

3.2.4 DER ZENTRALE KATALOG

Folgende Aspekte des Betriebs müssen in einer zentralen Datenstruktur erfaßt werden, um einerseits die Rekonstruktion der Arbeitsumgebung nach Unterbrechungen (geplant oder ungeplant) zu ermöglichen, und andererseits den Wechsel der Benutzer von einer Station an eine andere zu erlauben:

- das *Layout der Schreibtischoberfläche*, das als Attribut des Benutzers erscheint,
- die *Zugriffsattribute* mit den Zugriffsrechten für Benutzer zu *Objekten* (Veränderung und Löschung) sowie für die Neuanlage und allgemeine Benutzung eines *Bearbeitungswerkzeugs* in Form eines *Objekttyps*,
- die *Zuordnung der Anwendungsprogramme* (Werkzeuge) zu Fensterkonfigurationen bzw. Piktogrammen,
- die für bestimmte Basisoperationen (Löschen, Ausdrucken) erforderlichen Informationen über die *vorliegende Konfiguration und Interne Repräsentation*.

Die hierfür benötigten Daten sind in einer zentralen Datenbank, genannt *GDM-Katalog*, abgelegt. Bei Beginn einer Sitzung unter GDM werden die für einen Benutzer relevanten Daten als Momentaufnahme aus der Datenbank abgefragt, so daß zur Laufzeit keine Wartezeiten entstehen. Änderungen werden je nach Verfügbarkeit des Datenbank-Servers frühestmöglich in die Datenbank eingespielt, so daß hinsichtlich des Schreibtisch-Layouts keine Konsistenzprobleme entstehen. Dies enthebt selbstverständlich keines der beteiligten Werkzeuge der Verpflichtung, für eigene Sicherungspunkte während des Ablaufs zu sorgen.

Im folgenden sind die benötigten Datenstrukturen kommentiert dargestellt. Die Schreibweise folgt der quasi-relationalen Notation, wobei jedoch Listen und Attributkomplexe (Strukturen) zugelassen werden. Als Übersicht zeigt Abb. 3-15 das Konstruktionsdiagramm für den GDM-Katalog (*).

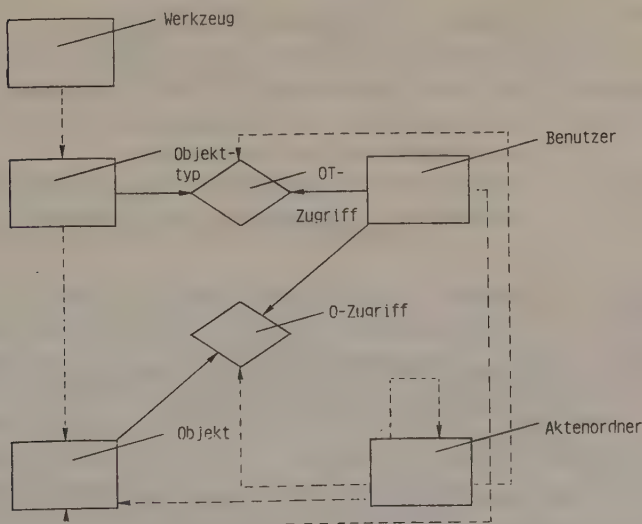


Abb. 3-15: GDM-Katalog - Konstruktionsdiagramm

(*) Das Diagramm folgt der Notation für Entity-Relationship-Diagramme nach WEDEKIND81 (S. 124f).

Schema:

werkzeug (w#, exe, aufrufmenü, aufrufpar)

Diese Struktur beschreibt Aspekte des Gebrauchs von Bearbeitungswerkzeugen im Rahmen von GDM.

<i>w#</i>	Eindeutige Kennzeichnung eines Werkzeugs
<i>exe</i>	Verweis auf das ausführbare Objekt des Werkzeugs
<i>aufrufmenü</i>	Optionalen Verweis auf das interpretierbare Objekt eines Menüs, das die möglichen Aufrufparameter entgegennimmt bzw. anbietet
<i>aufrufpar</i>	Template für die Kommandozeile des Aufrufs unter Einbezug von eventuellen Default-Parametern

Schema:

objekttyp (ot#, piktogramm, beschriftung, o_piktogramm, ass_werkzeug, int_rep, druck_treiber, def_part)

Hier wird der systemseitig benutzte Begriff *Objekttyp* an die Systemumgebung und Anwendungsprogramme angepaßt.

<i>ot#</i>	Eindeutige Kennzeichnung eines Objekttyps
<i>piktogramm</i>	Rasterisierte Darstellung der Ikone, die auf der Schreibtischoberfläche den Objekttyp symbolisiert
<i>beschriftung</i>	Beschriftung der Objekttyp-Ikone
<i>o_piktogramm</i>	Rasterisierte Darstellung des Piktogramms für die Darstellung von Objekten als Instanzen des Objekttyps
<i>ass_werkzeug</i>	Verweis auf das mit dem Objekttyp assoziierte Bearbeitungswerkzeug (Wertebereich <i>w#</i>)
<i>int_rep</i>	Beschreibung der Internen Repräsentation einer Instanz des Objekttyps; falls diese Attribut leer

ist, wird ein *transienter Objekttyp* angenommen (z.B. eine Terminalemulation)

druck_treiber Verweis auf eine für Instanzen des Objekttyps gültige Gruppe von Drucktreibern (im Falle eines *permanenten Objekttyps*)

def_part Beschreibung der Anfangskonfiguration für das einem Objekt des Objekttyps zugeordnete Fenster mit der Bemaßung der Partitionen und der Benennung der assoziierten Präsentationsflächentreiber.

Schema:

benutzer (b#, name, kurzname, stammknoten)

Diese Datenstruktur beschreibt einen GDM-Benutzer und bietet mnemonische Hilfen zu seiner Identifikation an, die auch von Werkzeugen, z.B. Kommunikationsprogrammen, genutzt werden können. Sämtliche benutzerspezifischen Daten werden über die Benutzerkennung in einzelne Kontexteinheiten separiert. Bei Wechsel eines Benutzers in eine Arbeitsplatzrechnergruppe, die seinen Stammknoten nicht enthält, kann so über Fernzugriff die gewohnte Arbeitsumgebung transferiert werden.

<i>b#</i>	Eindeutige Kennzeichnung eines Benutzers
<i>name</i>	Voller Name des Benutzers
<i>kurzname</i>	Kurzname des Benutzers
<i>stammknoten</i>	Verweis auf den Datenserver, der die Umgebung des Benutzers verwaltet

Schema:

aktenordner (a#, beschriftung, plazierung)

Im Rahmen von GDM werden durch ein entsprechendes Piktogramm symbolisierte Aktenordner als, ggf. auch rekursiv einsetzbares, Strukturierungs- und Ablageinstrument verwendet.

<i>a#</i>	Eindeutige Kennzeichnung eines Aktenordners
<i>beschriftung</i>	Name des Aktenordners (vom Benutzer vergeben)
<i>plazierung</i>	Koordinatenangabe bezogen auf die Virtuelle Schreibtischoberfläche bzw. auf einen übergeordneten Aktenordner, dessen Kennung Teil der Koordinatenangabe ist

Schema:

ot_zugriff (ot#, b#, plazierung)

Diese Datenstruktur legt fest, welcher Benutzer auf welchen Objekttyp bzw. welches Bearbeitungswerkzeug zugreifen darf. Zugriffsberechtigungen werden dabei durch die Sichtbarkeit auf der individuellen Schreibtischoberfläche ausgedrückt.

<i>ot#</i>	Eindeutige Kennzeichnung des betroffenen Objekttyps (Wertebereich <i>ot#</i>)
<i>u#</i>	Eindeutige Kennzeichnung des betroffenen Benutzers (Wertebereich <i>b#</i>)
<i>plazierung</i>	Koordinatenangabe bezogen auf die Virtuelle Schreibtischoberfläche bzw. auf einen übergeordneten Aktenordner, dessen Kennung Teil der Koordinatenangabe ist

Schema:

*objekt (o#, eigentümer_b#, ot#, beschriftung,
int_rep, plazierung)*

Diese Datenstruktur verwaltet Zugriffsinformationen und Layout-Informationen zu permanenten Objekten.

<i>o#</i>	Eindeutige Kennzeichnung des Objekts
<i>eigentümer_b#</i>	Verweis auf den Eigentümer des Objekts (Wertebereich <i>b#</i>)
<i>ot#</i>	Verweis auf den Objekttyp, dem das Objekt angehört (Wertebereich <i>ot#</i>)
<i>beschriftung</i>	Name des Objekts (vom Benutzer vergeben)
<i>int_rep</i>	Verweis auf Strukturen des Internen Repräsentation gemäß den in <i>objekttyp</i> fixierten Konventionen
<i>plazierung</i>	Koordinatenangabe bezogen auf die Virtuelle Schreibtischoberfläche bzw. auf einen übergeordneten Aktenordner, dessen Kennung Teil der Koordinatenangabe ist

Schema:

o_zugriff (o#, u#, plazierung)

Neben der standardmäßigen Zuweisung des Zugriffsrechts an den Eigentümer können hier noch weitere Zugriffsrechte festgehalten werden. Dabei sind jedoch auch benutzerspezifische Layout-Vorschriften zu beachten.

<i>o#</i>	Eindeutige Kennzeichnung des betroffenen Objekts (Wertebereich <i>o#</i>)
<i>u#</i>	Eindeutige Kennzeichnung des betroffenen Benutzers (Wertebereich <i>b#</i>)
<i>plazierung</i>	Koordinatenangabe bezogen auf die Virtuelle Schreibtischoberfläche bzw. auf einen übergeordneten Aktenordner, dessen Kennung Teil der Koordinatenangabe ist

3.2.5 DIE ANWENDUNGSPROGRAMMSCHNITTSTELLE

Konventionelle & GDM-basierte Bearbeitungswerkzeuge. Wie bereits bei den Anforderungen aufgeführt, können prinzipiell zwei Klassen von Bearbeitungswerkzeugen unterschieden werden:

- (i) *Konventionelle Bearbeitungswerkzeuge* stützen sich auf die Annahme, daß ein herkömmliches Datenendgerät als Ausgabemedium dient. Die Funktion von Datenendgeräten ist durch eine Anzahl von Betriebs- und Interpretationsarten für die Ausgabe der übertragenen Information charakterisiert. In diesem Fall muß das erwartete physische Datenendgerät durch ein transientes GDM-Objekt substituiert werden. Gleichermäßen kann bei Bearbeitungswerkzeugen, die Ein-/Ausgabeoperationen über mehrere Endgeräte ausführen, der Partitionsmechanismus hierfür eingesetzt werden. Ein konventionelles Bearbeitungswerkzeug hat in der Regel eine eigene Dialog- und Kommandoverwaltung, die innerhalb der zu einem gegebenen Zeitpunkt aktiven Partition präsentiert wird. Die Portierung eines solchen Werkzeugs erfordert einerseits die Bereitstellung eines Präsentationsflächentreibers, der die Funktionen des erwarteten Endgeräts emuliert, andererseits einen erneuten Bindevorgang mit den von diesem Treiber bereitgestellten Prozeduren; im Normalfall wird die Steuerung allerdings durch eingestreute Fluchtsymbol-Sequenzen in der auszugebenden Information erfolgen.
- (ii) *GDM-basierte Bearbeitungswerkzeuge* nutzen die von GDM angebotene Hilfsmittel zur Ausgabe- und Kommandomenüverwaltung aus. Dabei kommen in erster Linie abstraktere Organisationsmodelle zum Zuge, die auch eine kompaktere Darstellung des Präsentationsflächeninhaltes erlauben.

Behandlung von Ein- & Ausgaben. Ein-/Ausgabeoperationen können grundsätzlich nur zwischen dem Benutzer und aktiven Objekten bzw. dem Globalen GDM-Verwaltungsprogramm erfolgen. Die durch ein geöffnetes Fenster symbolisierte Aktivität bezieht sich auf den zugeordneten Applikationsprozeß, dem die Präsentationsflächen seines Fensters zu jedem Zeitpunkt für Ausgabeaktivitäten zur Verfügung stehen. Da im Normalfall jedoch nur ein Satz an Eingabegeräten (Tastatur, Maus) zur Verfügung steht, muß dieser Datenstrom an eine hierfür ausgezeichnete Präsentationsfläche-cum-Applikationsprozeß geleitet werden (Fachterminus: *listener*). Es bleibt dem dort aktiven Präsentationsflächentreiber überlassen, für eine der Eingabe entsprechende Reaktion zu sorgen (Echo) und Zeigeeinstrumenteingaben mit den entsprechenden Funktionstasten zu interpretieren. Der Wechsel des Empfängerprozesses wird durch ein Selektieren eines anderen Fensters bewirkt, woraufhin GDM durch eine Veränderung der Überlagerungsstruktur das neue Fenster über die vorhandenen überlagert sowie ggf. alle weiteren Kontextinformationen (z.B. Änderung der Darstellung der Verfolgungsmarke für die Maus) verändert. Weitere vom Anwendungsprogramm (z.B. einem Textverarbeitungswerkzeug) verlangte Schreibmarken müssen von diesem selbst oder dem zugeordneten Treiber verwaltet werden.

Eine von einem Anwendungsprogramm initiierte Eingabefunktion liest entweder aufgeforderte oder unaufgeforderte Eingaben, im zweiten Fall von einer Eingabewarteschlange, einem Mechanismus, der je nach Betriebsart des Anwendungsprogramms aktiviert oder deaktiviert sein kann. Während bei unaufgeforderten Eingaben der Bezug zum Inhalt einer Präsentationsfläche *manuell* durch Navigation mit den Blätterwerkzeugen hergestellt werden muß, kann bei expliziter Anforderung einer Eingabe eine Koordinatengabe des Bezugspunktes mitgeliefert und so der Bezug automatisch durch implizites Blättern hergestellt werden, falls der bezeichnete Punkt nicht sichtbar ist. Zur Abstimmung der Sichtbarkeit der Externen Repräsentation von Information in *verschiedenen* Präsentationsflächen können die zu einem gegebenen

Zeitpunkt vorliegenden Sichtbarkeitsverhältnisse durch das Anwendungsprogramm abgefragt werden.

Menüauswahl. Bei in hohem Maße interaktiven Systemen haben sich die menüorientierten (sprich: auswahlorientierten) Verfahren gegenüber den sprachorientierten Ansätzen durchgesetzt; dies liegt einerseits in der leichteren Bedienung auch für Anfänger, andererseits jedoch in der weitaus geringeren Fehlerhäufigkeit durch Syntaxfehler begründet. GDM bietet für GDM-basierte Werkzeuge Unterstützung nach den im folgenden beschriebenen Prinzipien an; gemäß der in Abschnitt 3.1.1 getroffenen Unterscheidung in N- und P-Handlungen sowie dem dort genannten Grundsatz der Präzedenz der N- vor der P-Handlung (Modusfreiheit) wird ein Kommando in folgende Grundeinheiten zerlegt:

- (Optionale) N-Handlung: Auf der Präsentationsfläche wird durch Selektion mit dem Zeigeeinstrument das zu bearbeitende Objekt angewählt; die korrekte Segmentierung der dargestellten Information obliegt dabei dem zuständigen Treiberprogramm, gleichermaßen die für die Rückkopplung erforderliche Hervorhebung des selektierten Objektes. KOVED86 schlägt eine analoge Vorgehensweise als *eingebettete Menüs* vor, wobei hier jedoch zwischen N- und P-Handlungen nur sehr unpräzise unterschieden worden ist.
- P-Handlung: Durch Auslösen eines Signals (meist: reservierte Funktionstaste auf dem Zeigeeinstrument) wird die für das selektierte Objekt gültige Funktionsauswahl sichtbar. Die dabei üblichen Techniken sind die

- Darstellung in einer Kommandoleiste an einer hierfür vorgesehenen Stelle des Bildschirms (Vorteil: *click ahead* möglich) und
- Darstellung in einem Pop-up-Menü, einem temporären Fenster, das nur in der Zeit der Kommandoausführung sichtbar ist (Vorteil: Freie Anordnung des Kommandofensters, meist direkt neben der Schreibmarke).

GDM benutzt letztere Technik und erlaubt dabei alternativ die Definition von Funktionstasten für häufig benutzte Kommandos.

- (Optionale) Parametrierung: Weiterhin ist die Definition von Parametrierungsmasken erlaubt, die nach Auslösen des Kommandos (nun allerdings modusbehaftet) eine weitere Parametrierung des Kommandos erlauben; auch hierfür wird die Pop-up-Technik verwendet.

Menüs werden nach Start eines Bearbeitungswerkzeugs deklariert, die Parametrierungsmasken werden dabei in interpretierbarer Form durch Verweis auf eine entsprechende Objektbibliothek beigelegt. Einzelne Menüeinträge können je nach Dialogstatus deaktiviert und wieder reaktiviert werden.

Die dargestellte Vorgehensweise auf der Ebene der Präsentationsflächen gilt analog für das Vorgehen auf Fensterebene, nur beziehen sich die hierfür definierten Kommandos nicht auf ein selektiertes Anwendungsobjekt, sondern auf die Anwendung insgesamt. Gleichermäßen sind von GDM selbst auch Kommandos auf der Virtuellen bzw. Physischen Schreibtischoberfläche definiert.

Präsentationsverwaltung. Manipulationen auf Desktopebene können die Neudarstellung von Fenstern oder Fensterinhalten erfordern. GDM verfolgt in diesem Zusammenhang eine duale Strategie: Verschiebungen und Verlagerungen resultieren in einem Wiederaufbau auf Fensterebene und können mit Hilfe der durch das

Fensterverwaltungsprogramm zurückbehaltenen Rasterdarstellung des Fensters rekonstruiert werden; eine Verschiebung auf Partitionsebene wird durch die im Treiberprogramm für die jeweilige Präsentationsfläche enthaltene logische Struktur des Präsentationsflächeninhalts inkrementell auf besagte Rasterdarstellung des Fensters abgebildet. In jedem Fall wird sichergestellt, daß keine derartige Anforderung bis auf Anwendungsebene vordringt.

Zur Orientierung unterstützt GDM eine Katalogstruktur, die alle zu einem gegebenen Zeitpunkt geöffneten Fenster bzw. auf der Schreibtischoberfläche vorhandenen Piktogramme per Namen in einer Liste zur Verfügung stellt. Nach Anwahl eines Namens wird der sichtbare Ausschnitt der Physischen Schreibtischoberfläche entsprechend angepaßt und ggf. die Präzedenz der Überlagerungsstruktur geändert, so daß das bezeichnete Objekt sichtbar wird. Die anderweitigen Positionierungs- bzw. Panningmechanismen der Physischen Schreibtischoberfläche bleiben davon unberührt.

3.2.6 ORGANISATIONSMODELLE UND INFORMATIONÜBERTRAGUNG

Dieser Abschnitt präzisiert den Einsatz von Organisationsmodellen im Zusammenhang mit der GDM Bildschirmverwaltungstechnik. Prinzipiell wird dieser Begriff analog zu seinem Gebrauch in Abschnitt 2.3 als Rahmenorganisation der Dokumentenarchitektur des Dokumentmonitors verwendet. Beispiele für Organisationsmodelle, die bei GDM zum Einsatz kommen, sind:

- das *Rastermodell*, das einfach Bildpunkte auf die Präsentationsfläche abbildet,

- das *Datenendgeratemodell*, das der Funktionsbeschreibung eines der de-facto-Standards auf diesem Markt folgt,
- das *kalligrafische Modell*, das aus grafischen Primitiven Segmente und letztendlich komplexe geometrische Objekte zusammensetzt,
- das *Formularmodell*, das Texteinheiten nach bestimmten Layoutvorschriften als Konstante bzw. Eingabetext behandelt,
- das *Dokumentmodell* nach einem der in Abschnitt 2.3 skizzierten Modelle
- sowie *anwendungsorientierte Modelle*, die beliebigen, vom Anwendungsprogrammierer auf die Anforderungen bestimmter Programmgruppen hin spezifizierten Funktionen und Darstellungsformen entsprechen.

Die folgenden Betrachtungen zur Informationsübertragung beruhen auf dem in Abschnitt 2.3 definierten Begriff der *Konstruktiven Abhängigkeit*, zu der ein Übersichtsdiagramm zu den genannten Datentypen in Abb. 3-16 wiedergegeben ist.

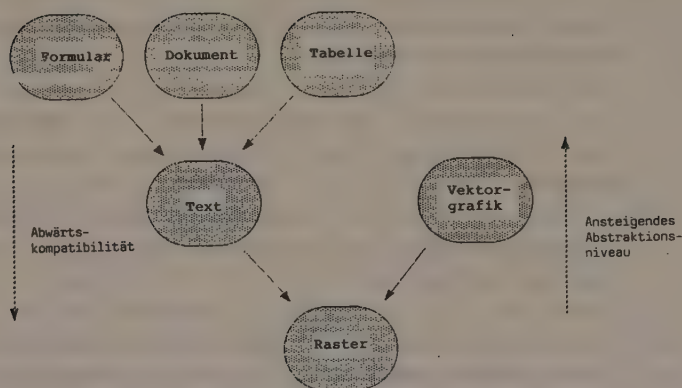


Abb. 3-16: Konstruktive Abhängigkeit von
Standard-Organisationsmodellen

Die in der Abbildung dargestellte Hierarchie gibt gleichzeitig die Pfade für einen Informationstransfer von einer Präsentationsfläche zu einer anderen an. Grundlage für diese Vorgehensweise ist die Forderung, daß jedes für ein Organisationsmodell vorhandene Treiberprogramm einen Konvertierungsdienst zu dem logischen Datenformat vorsieht, von dessen Organisationsmodell es konstruktiv-abhängig ist. Bei einem Transfer werden nun aufgrund der jeweils betroffenen Fensterkonfigurationen die beteiligten Organisationsmodelle der Sender- bzw. Empfänger-Präsentationsfläche ermittelt. Der Transfer kann nun

auf der Basis des Organisationsmodells stattfinden, das die Wurzel des kleinsten gemeinsamen Teilbaums hinsichtlich der Hierarchie der konstruktiven Abhängigkeit bildet. Falls das so gewählte Übertragungsmodell nicht mit dem Empfängermodell identisch ist, muß der Empfänger entweder die Weiterbearbeitung auf der Basis des weniger abstrakten Übertragungsmodells weiterführen oder (soweit möglich) für eine Ergänzung bzw. Re-Interpretation der übertragenen Information sorgen.

Beispiel: Bei der Übertragung von Information aus einem Organisationsmodell zur Bearbeitung dreidimensionaler geometrischer Objekte (mit impliziter Projektion auf einen gegebenen Betrachterstandpunkt) in eine Präsentationsfläche mit zweidimensionalem geometrischem Modell wird der Transfer nach der Parametrierung der Konvertierungsfunktion durch Projektionsparameter auf zweidimensionaler Ebene stattfinden. Eine umgekehrte Transferrichtung ist nur schwer vorstellbar, da im allgemeinen Fall die Rekonstruktion dreidimensionaler Koordinaten aus einer zweidimensionalen, projizierten Darstellung nicht möglich ist.

Das Transfermedium ist in jedem Fall ein uninterpretierter Pufferbereich, der über einen Namen (clipboard) referenziert wird.

Die konstruktive Abhängigkeit von Organisationsmodellen ist Gegenstand einer Erweiterung des GDM-Katalogs, die im folgenden dargestellt ist. Abb. 3-17 zeigt das zugehörige Konstruktionsdiagramm.

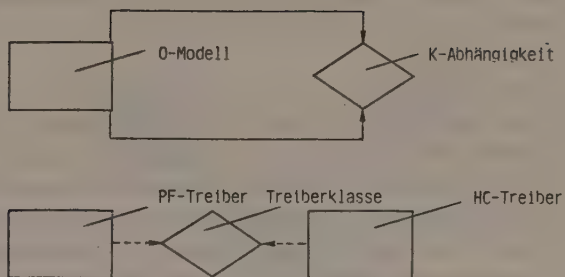


Abb. 3-17: GDM-Katalog (Erweiterung) - Konstruktionsdiagramm

Schema:

`o_modell (om#, name, treiberklasse)`

Diese Struktur beschreibt die verwendeten Organisationsmodelle.

<code>om#</code>	Eindeutige Kennzeichnung des Organisationsmodells
<code>name</code>	Name des Organisationsmodells
<code>treiberklasse</code>	Verweis auf die Treiber, die für verschiedene Datenendgeräte das Organisationsmodell implementieren

Schema:

k_abhängigkeit (*von_om#*, *zu_om#*)

Diese Struktur beschreibt die Hierarchie bzgl. der konstruktiven Abhängigkeit.

<i>von_om#</i>	Eindeutige	Kennzeichnung	des
	Organisationsmodells,	von dem <i>zu_om#</i>	abhängig ist
<i>zu_om#</i>	Eindeutige	Kennzeichnung	des
	Organisationsmodells,	das von <i>von_om#</i>	abhängig ist

Schema:

pf_treiber (*pft#*, *zielgerättyp*, *klasse*)

Diese Struktur beschreibt Präsentationsflächentreiber und die Zordnung zu den verschiedenen Typen von Datenendgeräten.

<i>pft#</i>	Eindeutige Kennzeichnung des Präsentationsflächentreibers
<i>zielgerättyp</i>	Bezeichnung des Datenendgerättyps
<i>klasse</i>	Zuordnung zu einer Treiberklasse

Schema:

hc_treiber (*hct#*, *zielgerättyp*, *klasse*)

Diese Struktur beschreibt Hardcopy-Treiber und die Zuordnung zu den verschiedenen Typen entsprechender Ausgabegeräte.

<i>hct#</i>	Eindeutige Kennzeichnung des Hardcopy-Treibers
<i>zielgerättyp</i>	Bezeichnung des Ausgabegerättyps
<i>klasse</i>	Zuordnung zu einer Treiberklasse

4 Büroprozeduren

Nach der Betrachtung primitiver Interaktionen werden in diesem Abschnitt Beschreibung und Ablauf komplexer Vorgänge das Thema sein. Bis zu diesem Punkt wurde der üblicherweise verwendete Begriff *Buroautomation* weitestgehend vermieden, da der Begriff *Automation*, der in dieser Zusammensetzung enthalten ist, weitraumige Abläufe im Sinne einer *Prozedur* impliziert. Die Prozeduren, die zur Konstitution von Organisationsmodellen dienen, sind rein programmtechnische Konstrukte und haben mit Prozeduren im Büro, d.h. Komplexen mehrerer ggf. parallel ablaufender Vorgänge, nichts gemein. Der allgemein verwendete Terminus ist *Büroprozedur*; es ist dabei also nur das Einsatzgebiet genannt, über die Eigenschaften ist a priori nichts bekannt. Als besondere Anforderungen dieses Gebietes an Prozedurmechanismen sind dabei anzuführen (vgl. auch HOGG85b, S.137f):

- (i) *Flexibilität*: Ein Bürovorgang kann nicht bis ins einzelne spezifiziert werden; im Normalfall sind entweder filigrane Fallunterscheidungen, abhängig vom jeweiligen Umfeld eines Vorgangs, zu treffen, oder es ist durch hinreichende Modularisierung und Dezentralisierung durch ein *Stepwise-Refinement-Verfahren* aus einem vagen Prototyp einer Prozedur die konkrete Instanz zu generieren, die dem Spezialfall entspricht.

- (ii) *Ausnahmebehandlung*: Büroprozeduren werden im allgemeinen nur unvollständige Angaben über die zur Laufzeit bestehende Umgebung haben; deshalb muß der Behandlung von Ausnahmesituationen breiter Raum gelassen werden. Einige Ansätze (vgl. u.a. KRATZER85, für den allgemeinen Fall aber auch BORGIDA85) sehen Bürotätigkeit sogar ausschließlich als Bearbeitung von Ausnahmesituationen an (*Mangelbehebung* und *Konfliktbeseitigung*).
- (iii) *Problemlösen*: Büroprozeduren sollen, um den Automatisierungsaspekt nicht zu verlieren, weitestgehend autonom arbeiten können. Zu diesem Ziel müssen zur Laufzeit die Zusammenhänge von Abläufen im Büro explizit gemacht werden, um Entscheidungen selbständig treffen oder doch mindestens kompetent vorbereiten zu können.
- (iv) *Lokalität*: Büroprozeduren werden gemeinhin nicht zentral erstellt und verwaltet; vielmehr sind Bearbeitungsvorgänge nur auf sehr abstrakter Ebene zu formulieren und dann im Bedarfsfall zu verfeinern.

In den folgenden Abschnitten soll der Begriff *Büroprozedur* in Analogie zum Begriff *Bürohandlung* rekonstruiert werden, wobei die im Büro gegebenen Randbedingungen berücksichtigt werden sollen. Ein weiterer Abschnitt befaßt sich mit den bereits eingeführten Modellen zur Beschreibung von Büroprozeduren. Zum Abschluß soll ein Abgleich des Begriffs *Büroprozedur* mit dem in der Datenbanktechnik eingeführten Begriff *Transaktion* die Probleme des Einsatzes von Datenbanksystemen im Bürobereich beleuchten.

4.1 HANDLUNGEN IM BÜRO

In einem Rückgriff auf die Handlungstheorie wird im folgenden versucht, Umfeld und Struktureigenschaften menschlicher oder maschineller Aktivität zu ergründen.

4.1.1 ZUM HANDLUNGSBEGRIFF

Definition. Vorweg sei eine naive, aber dennoch anerkannte Definition aus dem Bereich der Handlungspsychologie genannt: Hacker beschreibt eine Handlung als die "kleinste psychologische Einheit der willensmäßig gesteuerten Tätigkeit" (HACKER78, S.62). Hilft diese Definition hinsichtlich einer Ergründung vor Büroaktivitäten auch kaum weiter, da sie ausschließlich auf atomare Aktivitätsquanten ohne Berücksichtigung des Umfelds zurückgreift, so ist damit jedoch bereits der Zusammenhang zwischen einer Handlung und der ihr zugrundeliegenden Intention ausgedrückt. Der dadurch geprägte Handlungsbegriff wird in der Wissenschaftstheorie als *intentionalistisch* bezeichnet, im Gegensatz zu der Betrachtungsweise der *Behavioralisten*, die beobachtbare Naturereignisse mit einbeziehen (vgl. hierzu KAMBARTEL80, S. 97), damit jedoch den hier kaum einbezieharen Bereich der Metaphysik berühren; dies wird ja schon sprachlich hinsichtlich der Auswirkungen von Naturereignissen als *Höhere Gewalt* (am.: Act of God) angedeutet.

Eine erste Eingrenzung der zu betrachtenden Handlungen ergibt sich aus dem Anwendungsbereich: nachdem im Büro nur symbolische Handlungen vorgenommen werden, die materiale Handlungen nachvollziehen oder vorwegnehmen (z.B. Wechselwirkung zwischen Verwaltungs- und Fertigungsbereich einer Fabrik), kann das Han-

deln im Büro auf *Sprechhandlungen* zurückgeführt werden. Derartige Handlungen können sich durchaus *materialer Handlungen* als *Trägerhandlungen* bedienen und wiederum als *Vermittlungshandlungen* für auszulösende *materiale Handlungen* angesehen werden (z.B. Ausfüllen eines Fertigungsauftrages). Die Standardform für *Sprechhandlungen* ist, in Anlehnung an Gethmann,

«*Performer*» «*Prädikator*» «*Nominator*»,

wobei der *Performer* die Handlung in eine der Kategorien des *Aufforderns*, *Behauptens*, *Zweifeln*s, *Zustimmens* oder *Bestreitens* einordnet (vgl. GETHMANN^{79a}, S. 85); *Prädikator* und *Nominator* werden entsprechend den bereits in Abschnitt 3.1.1 eingeführten *P-* und *N-Handlungen* eingesetzt.

Ähnlich wie im Falle von Datenstrukturen lassen sich *Handlungen* auch über eine Typisierungsoperation klassifizieren, d.h., für eine Einzelhandlung kann durch Zuspreehen bestimmter *Eigenschaften* die Zuordnung zu einem Typ konstatiert werden, der diese *Eigenschaften* fordert (vgl. auch STEINBAUER⁸³, S. 37).

Die Zusammensetzung einer Handlung wird wiederum als Handlung aufgefaßt, die die "Klammerfunktion" hinsichtlich der *Elementarhandlungen* erfüllen soll. Das programmtechnische Analogon hierfür ist die *Verbundanweisung* bzw. die *Kontrollstruktur* im allgemeinen.

Das Handlungsschema. Unter einem Handlungsschema versteht man die Sammlung der Merkmale, die einen Handlungstyp bestimmen (vgl. STEINBAUER⁸³, S. 39). Damit kann der Typ einer einzelnen Handlung über ein solches Schema bestimmt werden, wenn die dem Schema inhärenten Handlungsmerkmale bestimmten *Eigenschaften* genügen. Für die Verwendung des Begriffs in einem prozeduralen Kontext sind folgende *Eigenschaften* bestimmend (vgl. hierzu STEINBAUER⁸³, S. 40ff):

- *Subjektklasse einer Handlung*: Jede Handlung kann einem Subjekt zugeordnet werden, das Träger der Intention ist, die letztlich die Handlung auslöst. Handlungsseitig ist dabei die *Befähigung* zur Ausführung der Handlung zu berücksichtigen, die zunächst von der *Zulässigkeit* der Ausführung der Handlung durch ein Subjekt abzugrenzen ist. Dieser Unterschied wird jedoch im Falle der ständigen Überwachung von Handlungen durch eine übergeordnete Instanz zumindest dahingehend egalisiert, daß die Befähigung zur Ausführung einer Handlung durch ein Subjekt sehr wohl durch die für diese Konstellation gültige Zulässigkeit begrenzt werden kann (Zugriffsschutzmechanismen, *Capabilities*), jedoch die umgekehrte Abgrenzung nur organisatorisch erfolgen kann, d.h. technisch nicht oder nur unpräzise (z.B. über das Kompetenzprofil eines Benutzers) erfaßt werden kann.

- *Objektklasse einer Handlung*: Jede Handlung manipuliert Objekte. Dabei sind die *unmittelbar betroffenen* Objekte, die im Zuge der Durchführung der Handlung entstehen, ihren Zustand verändern oder vernichtet werden, von den *mittelbar betroffenen* Objekten zu unterscheiden, die entweder als *Vermittlungsobjekte* dienen oder den Ablauf der Handlung durch ihren Zustand beeinflussen. Im ersten Fall ist die Zugriffsart *schreibend*, im zweiten Fall *lesend*. Ähnlich wie bei der Subjektklasse können hier prinzipiell auch Befähigungs- und Zulässigkeitskategorien unterschieden werden, diesmal jedoch hinsichtlich der Handlungen bezüglich der betroffenen Objekte. Da jedoch die Objekte in ihrem Verhalten erst durch die auf ihnen definierten Elementarhandlungen konstituiert werden, können Befähigung und Zulässigkeit in diesem Falle als synonym angesehen werden.

- *Handlungssituation*: Eine Handlung bezieht sich auf bestimmte Eigenschaften ihrer Umgebung in der Anwendungswelt; charakterisierend sind die Umgebungszustände vor und nach Ausführung der Handlung, ggf. auch während des Ablaufs der Handlung. Analog hierzu sind die in der Programmierung, namentlich im Verifikationsbereich, üblichen Vor- und Nachbedingungen zu sehen. Während des Handlungsablaufs entstehende Zwischenzustände können Rückwirkungen auf parallel ablaufende Handlungen haben (in Handlung A unmittelbar betroffene Objekte, die mittelbar in die Objektklasse einer parallel ablaufenden Handlung B eingehen).

Eine bedeutende Einschränkung der Handlungskontrolle stellt dabei die meist nicht sehr umfangreiche Einflusssphäre des Trägersystems dar. Wenn man von Sprechhandlungen im oben definierten Sinn spricht, muß man zwischen

- *Intersystem-Handlungen*, die Auswirkungen über das auslösende System hinaus haben können, und
- *Intrasystem-Handlungen*, die in ihren Auswirkungen innerhalb des auslösenden Systems beschränkt sind,

unterscheiden. Intrasystem-Handlungen sind problemlos kompensierbar, solange sie nicht innerhalb des Systems in materiale Handlungen umgesetzt worden sind; dieser Zeitpunkt ist auch ohne weiteres ermittelbar. Im Fall der Intersystem-Handlungen ist die Handlung dennoch sofort als material anzusehen, da der Übergangszeitpunkt hier gemeinhin nicht ermittelbar ist und so eine sofortige Auslösung einer vermittelten materialen Handlung angenommen werden muß.

- *Ablaufschema einer Handlung*: Handlungen ergeben sich allgemein aus rekursiver Konstitution aus Teilhandlungen bzw. Elementarhandlungen. Die Art und strukturelle Eigenschaft dieses Konstitutionsprozesses ist charakteristisch für den Typ der Handlung.

Für die Zuordnung einer konkreten Handlung zu einem Typ gilt also, daß die Handlung entsprechend den genannten Beurteilungskriterien die Eigenschaften erfüllen muß, die einem der Handlungsschemata entsprechen, die für den Typ charakteristisch sind. Für den Fall der automatisierten Erzeugung von Handlungen in einer Büroumgebung ist diese Betrachtungsweise natürlich dahingehend zu ergänzen, daß es hier weniger auf eine Zuordnung denn auf eine *Instantiierung* einer Handlung aus einem gegebenen Handlungsschema ankommt, hier also in der Regel zwischen Handlungsschemata und Handlungstypen eine 1:1-Beziehung herrscht.

4.1.2 HANDLUNGSVORSCHRIFTEN

Dieser Abschnitt versucht, die Typisierung von Handlungen dahingehend weiter zu konkretisieren, daß nun die *Intention* im Zusammenhang mit der Auslösung eines Handlungsablaufs betrachtet wird. Das primäre Unterscheidungskriterium ist hierfür die *Finalität* bzw. *Afinalität* der Formulierung der *Handlungsvorschrift*, wobei der letztgenannte Begriff als die Summe aus den oben genannten Beschreibungskriterien für Handlungen anzusehen ist.

4.1.2.1 AFINALE HANDLUNGSVORSCHRIFTEN

Eine *afinale Handlungsvorschrift* basiert auf der festen Vorgabe eines Handlungsschemas, das zur Instantiierung der Handlung dient. Die Auswirkungen der Handlungsvorschrift sind dabei im Detail bekannt. Die ausführende Instanz benötigt dafür folgende Informationen:

- den Namen des zugrundeliegenden Handlungsschemas (Prozedurname),
- die Kennzeichnung des für die Handlung gültigen Subjektes (Zugriffsschutz) sowie
- eine Liste zur Ergänzung der Objektklasse (außer Konstante), die der Handlung unterworfen ist (Parameter).

Die Formulierung einer afinalen Handlungsvorschrift im Büro erfordert also zu deren Rechtfertigung das Wissen um die zur Zeit der Instantiierung der Handlung vorliegenden Vorbedingungen und muß inhärent die Gewährleistung der Erfüllung der Nachbedingungen sicherstellen. Diese Bedingungen sind wohl kaum in der Umgebung eines Stabsbüros, wohl eher für Routinetätigkeiten in Linienbüros anzutreffen. BARBER82a (S. 345) sieht diese - in seiner Terminologie *prozeduralen* - Beschreibungsformen schon deshalb als ungeeignet an, da die vielen Randbedingungen und alternativen Vorgehensweisen, die im Büro (in diesem Fall dem Stabsbüro) zu beachten sind, auf diese Weise nicht erfaßt werden können.

4.1.2.2 FINALE HANDLUNGSVORSCHRIFTEN - DAS PROBLEMLÖSUNGSPARADIGMA

Im Gegensatz dazu sind *finale Handlungsvorschriften* durch die Vorgabe einer anzustrebenden *Nachbedingung* charakterisiert. Die ausführende Instanz einer derartigen Vorschrift verfügt über folgende Informationen:

- die Beschreibung der anzustrebenden *Nachbedingung* (Zielvorgabe),
- die Beschreibung der vorliegenden *Vorbedingung* zur Laufzeit (Eigenermittlung oder Übergabe bei Initiierung der Handlung),
- die Beschreibung der zur Verfügung stehenden *Elementaraktionen*.

Nachdem keine im Detail auszuführende Handlungsvorschrift vorgegeben ist, eignen sich Spezifikationen dieser Art für den Einsatz im Stabsbüro. Die Anforderungen der Flexibilität und der Adaptivität für Ablaufbeschreibungen wird durch die Abhängigkeit des Handlungsablaufs von den Vorbedingungen gewährleistet.

Was sind die technischen Implikationen dieses Ansatzes? In erster Linie sind zwei grundsätzliche Fragen zu beantworten: das *Ob* und das *Wie*. Dabei stellen sich folgende Probleme: Die *Erreichbarkeit* eines Nachzustandes und das *Erreichen* dieses Nachzustandes.

vertikalen Pfeile angedeuteten Implikationen den Zusammenhang zwischen ontischen und deontischen Modalitäten sicher. Die derart fixierten praktischen Modalitäten der *Unvermeidbarkeit* und der *Erreichbarkeit* sind für die weiteren Betrachtungen in erster Linie hinsichtlich der ihrer Beziehung zur Modalität der *Möglichkeit* von Bedeutung. *Möglich* ist, was aufgrund von Naturgesetzen vorstellbar ist, während die *Erreichbarkeit* auch die Angabe eines Weges, d.h. einer technischen Realisierung zur Erreichung eines Endzustandes, vorsieht.

4.1.2.2.2 PROBLEME UND PROBLEMLÖSEN

Zur Klassifikation von Problemen. Im folgenden wird ein Problem im Sinne von WEDEKIND⁸⁶ (S. 73) als Übergang von der theoretischen Modalität des Möglichen zu der des Erreichbaren verstanden. Dieser Übergang wird bei Dörner (DÖRNER⁷⁹, S. 10) als Barriere angesehen, die die Transformation eines unerwünschten Anfangszustandes in einen erwünschten Endzustand verhindert. Eine Klassifikation von Problemen nach dem vorliegenden Barrieretyp zeigt Abb. 4-2.

		Klarheit der Zielkriterien	
		hoch	gering
Bekanntheitsgrad der Mittel	hoch	Interpolations- barriere	dialektische Barriere
	gering	Synthese- Barriere	dialektische Barriere und Synthesebarriere

Abb. 4-2: Klassifikationsschema für Probleme
(aus DÖRNER⁷⁹, S. 14)

Die zweidimensionale Anordnung ergibt sich aus den beiden Einteilungskriterien *Bekanntheitsgrad der Mittel* und *Klarheit der Zielkriterien*. Letztere ist namentlich bei Alltagsproblemen oft nicht gegeben: Die Möblierung eines Raumes kann nach ästhetischen Gesichtspunkten kaum absolut bewertet werden, sondern nur im Vergleich zu alternativen Einrichtungsstilen (*Komparativkriterien*). Dörner bezeichnet diese Problemklasse als *dialektische Probleme*, da Lösungen nur im Vergleich von Alternativlösungen bzw. auf der Suche nach Widersprüchen ermittelt und ggf. verändert werden können.

Beim Bekanntheitsgrad der Mittel unterscheidet man zwischen *interpolativ-lösbaren Problemen* und *synthetischen Problemen*. Erstere sind durch die Bekanntheit der Übergänge, d.h. der jeweils für einen Transformationsschritt erforderlichen Werkzeuge, gekennzeichnet. Daraus folgt, daß nur die *Reihenfolge der Anwendung* der Zustandsübergänge zu ermitteln ist. Im Synthesefall ist das Operatorinventar offen und bildet durch seine Leerstellen ggf. Unterprobleme des Hauptproblems.

Welche Probleme eignen sich nun zur Bearbeitung durch Informationssysteme? Grundsätzlich können dialektische Probleme nicht direkt behandelt werden, sondern nur über eine wohldefinierte Abbildung in interpolativ-lösbare bzw. rein-synthetische Probleme übergeführt werden. Im betriebswirtschaftlichen Bereich spricht man dann von der Ermittlung einer *Figure of Merit*, an die sich eine Bearbeitung mit klarer Zieldefinition anschließen kann, im Bereich des Software Engineering werden durch derartige Präzisierungen *Aufgaben* in *Aufgabenstellungen* übergeführt.

Der Fall der synthetischen Probleme liegt etwas schwieriger: Man muß daran nämlich die Frage knüpfen, inwieweit das Phänomen der *Kreativität* der maschinellen Bearbeitung zugeführt werden kann. In seinen Ausführungen zu diesem Thema beschreibt Dörner einige Versuche, bei denen Versuchspersonen synthetische Probleme, die landläufig dem Bereich der Denksportaufgaben zuzurechnen sind,

lösen sollten. Entscheidend für eine erfolgreiche Bearbeitung war dabei die Fähigkeit, sich nicht von impliziten Einschränkungen des Lösungsraums beeinflussen zu lassen, sondern durch wörtliche Auslegung der Aufgabenbeschreibung neue Lösungsräume mit neuen Operatoren "aufzustoßen". Wenn man dieser Auslegung des Begriffs Kreativität folgt, so sind synthetische Aufgabenstellungen automatisch nicht bearbeitbar. Allerdings stellt sich in diesem Zusammenhang immer die Frage nach der Vollständigkeit der vorgegebenen Weltsicht und des Operatoreninventars. Bei unvollständigem Operatoreninventar ist immer die Möglichkeit gegeben, durch eine Vervollständigung eine a priori synthetische Aufgabe in den interpolativen Bereich überzuführen. Auch das synthetische Problemlösen durch Einbezug von Lösungswegen aus anderen Realitätsbereichen ist nur auf eine Unvollständigkeit der vorgegebenen Weltsicht zurückzuführen. Dies ist auch der Grund, weswegen gewissenhaftes menschliches Problemlösen seinem maschinellen Pendant vorerst immer überlegen sein wird: Die Überlegenheit des Menschen liegt in der Tatsache begründet, daß bis zu seiner Entscheidungsreife 20-25 Jahre der Erziehung, sprich: Programmierung, liegen, während Maschinen speziell und isoliert auf meist nur einen Aufgabenbereich zugeschnitten werden. Eines der Kernprobleme der Programmierung liegt in der sinnvollen Abgrenzung des zu bearbeitenden Themenbereichs, die jedoch bei Einbezug der Bearbeitung synthetischer Probleme nicht möglich ist.

Abschließend ist zu bemerken:

Maschinell sind nur interpolativ-lösbare Probleme zu bearbeiten; alle anderen Problemtypen müssen ggf. auf den interpolativen Fall zurückgeführt werden. Ist dies nicht möglich, muß von dem scharf formulierten Begriff der Problemlösungsautomatisierung auf den der Problemlösungsunterstützung übergegangen werden.

Interpolative Problemlösungsstrategien. Da in vielen Fällen eine einzige Anwendung eines Operators nicht ausreicht, ein Problem interpolativ zu lösen, müssen speziell entwickelte Strategien zur Ermittlung bzw. Erreichung eines gewünschten Endzustandes angewendet werden. DÖRNER⁷⁹ (S. 60ff) unterscheidet hierbei fünf disjunkte geistige Operationen zur Steuerung des Problemlösungsprozesses:

- *Situations- und Zielanalyse:* Die Situationsanalyse dient im Prozeß der Problemlösung der genauen Ermittlung der Eigenschaften der vorliegenden Ausgangssituation. Analog wird eine Analyse der anzustrebenden Zielsituation durchgeführt. Diese beiden Analysevorgänge dienen einerseits der Ermittlung der Unterschiede zwischen beiden Situationen, um daraus die einzuleitenden Maßnahmen ableiten zu können, andererseits der Feststellung des Endzustandes, wenn eine Situation mit den angestrebten Eigenschaften vorliegt. Nach ihrem Abschluß liegt eine detaillierte Liste der Anforderungen an den Problemlösungsprozeß vor.
- *Operatorauswahl:* Die Operatorauswahl aus einem vorgegebenen Inventar betrifft die Suche nach Mitteln, um die gegebene in die gesuchte Situation zu transformieren. Die Auswahl richtet sich dabei nach der bekannten Wirkung der Einzeloperatoren. Besondere Beachtung muß dabei die Analyse der bekannten Nebenwirkungen von Operatoren finden, gleichermaßen auch die Feststellung, ob durch die Wirkung eines Operators evtl. einer der Vorbedingungen eines später anzuwendenden Operators (im Sinne eines Handlungsschemas; vgl. Definition in Abschnitt 4.1.2) zuwiderläuft. Abb. 4-3 zeigt das Flußdiagramm eines Verfahrens zur Operatorauswahl; *D* bezeichnet dabei die Menge der Absichten, d.h. der zu beseitigenden Unterschiede zwischen Ausgangs- und Zielsituation, während *DE* die Menge

der aufgrund der anwendbaren Operatoren zunächst nicht erfüllbaren Absichten enthält.

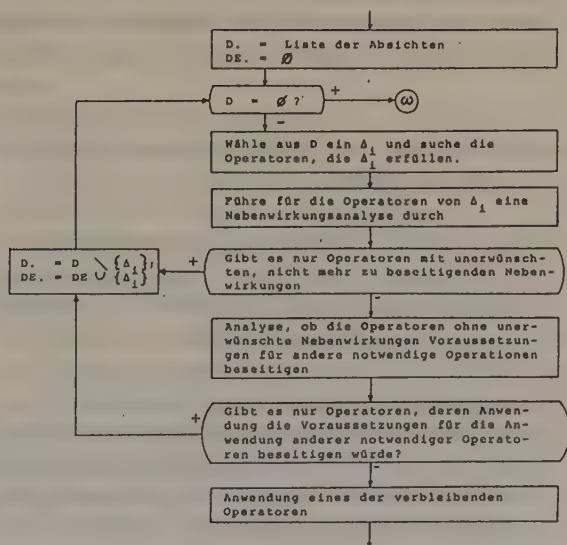


Abb. 4-3: Verfahren zur Operatorauswahl bei inter-polativen Problemen (aus DÖRNER⁷⁹, S. 63)

Allgemein gesehen zählt sich dabei die Einhaltung der Regel aus, zunächst Operatoren mit breitem Wirkungsspektrum einzusetzen, um schließlich durch Operatoren mit schmalere Wirkungsspektrum noch Detailkorrekturen durchzuführen.

- *Operatoranwendung und Erfolgsanalyse:* Nach der Auswahl erfolgt die Anwendung des Operators, falls Grund zu der Annahme besteht, daß dadurch eine Hinorientierung zum gewünschten Endzustand erreicht werden kann. Die Anwendung besteht aus zwei Teilprozessen, der
 - *Prüfung der Anwendbarkeit* des Operators und der
 - *Ausführung der Teiloperationen*, aus denen ein Operator besteht.

Falls nach der Prüfung die Anwendbarkeit nicht festgestellt werden kann, ist eine *Zwischenzielbildung* im Sinne der Formulierung eines Teilproblems der eigentlichen Ausführung vorzuschalten.

Das eigentliche Problem stellt sich erst im Zuge der auf die Operatoranwendung folgenden *Erfolgsanalyse*. Was ist Erfolg? Um Erfolg meßbar zu machen, muß über eine Bewertungsfunktion der Unterschied zweier Situationen, im genannten Fall natürlich der Abstand einer vorliegenden Situation von der Zielsituation, numerisch erfaßbar gemacht werden. Derartige Bewertungsskalen können durchaus mehrdimensional angelegt werden und über eine weitere übergeordnete Zielfunktion auf einer summarischen Bewertungsskala zusammengefaßt werden. Dabei treten die üblichen Optimierungsprobleme (Orientierungsproblem auf einem Plateau, Suboptima) auf, so daß in vielen Fällen zusätzliches Wissen die Bewertungsfunktion ergänzen muß. Allgemein wird natürlich der eigentliche Operator nicht angewendet werden, sondern nur mit seinen Eigenschaften in einen Planungsprozeß Eingang finden, der im Grunde eine simulative Anwendung einer Ausführungsfolge von Operatoren darstellt.

- *Umorientierung bei Mißerfolg:* Dem Scheitern des Versuchs der Anwendung eines Operators können entweder die *Nichtanwendbarkeit* oder die *Erfolglosigkeit* des Operators zugrundeliegen. Die daraufhin erforderlichen Umorientierungen können folgendermaßen klassifiziert werden:
 - *Zwischenzielbildung:* Diese ist an die Nichtanwendbarkeit des betrachteten Operators gebunden; wie bereits erwähnt, bildet sie den Ausweg, um einen nichtanwendbaren Operator dennoch anwenden zu können.
 - *Anwendung a priori mindergeeigneter Operatoren:* Nachdem die Frage der Anwendbarkeit bei der Operatorenauswahl bereits geklärt worden ist, kann bei Erfolglosigkeit ein mindergeeigneter Operator herangezogen werden, in der Hoffnung, daß einer seiner Nebeneffekte die Voraussetzung für effizientere Operatoren schafft.
 - *Absichtswechsel:* Dieser Schritt ist mit einem Wechsel der Merkmale der angestrebten Zielsituation gleichzusetzen; durch eine derartige Verschiebung können sich neue Wege der Lösungsfindung öffnen.
 - *Zielwechsel:* Durch Wechsel bzw. Re-Priorisierung der anstehenden Zwischenziele kann ein neuer Blickwinkel ein neues Unter-Inventar anwendbarer Operatoren schaffen.
 - *Startpunktwechsel:* Durch Rücksetzen der vorliegenden Situation auf den ursprünglichen Startpunkt oder ein zurückliegendes Zwischenziel können früher getroffene Fehlentscheidungen zurückgenommen werden. Dieser Ausweg ist im Bereich der Künstlichen Intelligenz als *Backtracking* bekanntgeworden; beispielsweise beruht das Ausführungsmodell der Programmiersprache PROLOG in hohem Maße auf dieser Technik (vgl. CLOCKSIN81, S. 58ff).

- **Wechsel des Heurismus:** Durch grundsätzliche Änderung der Vorgehensweise, z.B. durch Übergehen von einem planvollen Vorgehen auf die *Trial&Error-Technik* ohne weitergehende Berücksichtigung von Randbedingungen, kann die Lösung eines Problems gleichermaßen vorangetrieben werden.

Dörner systematisiert die Anwendung dieser Auswege durch das in Abb. 4-4 dargestellte Diagramm.

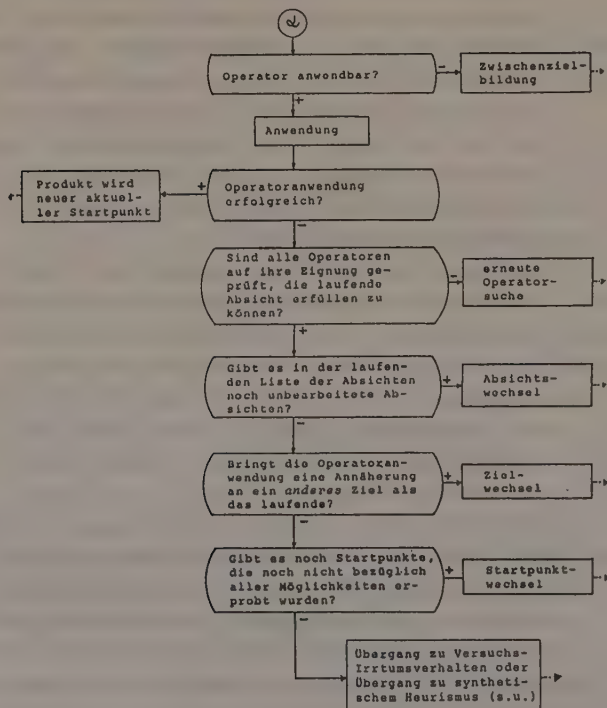


Abb. 4-4: Weiterverarbeitung bei Mißerfolg
(aus DÖRNER79, S. 68)

- *Festlegung der Suchrichtung:* Für die Ermittlung einer Anwendungsfolge für Operatoren bei gegebenem Ist- und Sollzustand stehen grundsätzlich zwei Verfahren zur Verfügung, die beide bereits implizit genannt wurden und analog zu den entsprechenden Verfahren in Begründungsdiskursen (vgl. Abschnitt 3.1.2) Anwendung finden:
- *Vorwärtsanwendung von Operatoren:* Sie wird entsprechend den bisherigen Betrachtungen ausgehend von einem Anfangszustand auf den Zielzustand geführt.
- *Rückwärtsanwendung von Operatoren:* Sie versucht vom Zielzustand durch Ersetzung der Konsequenzen eines Operatoreinsatzes durch die einschlägigen Prämissen den Ausgangszustand zu erreichen.

Abb. 4-5 zeigt den zwischen Anfangs- und Zielzustand durch das Operatoreninventar aufgespannten Raum an Zwischenlösungen und den entsprechenden Einsatz der Basisstrategien.

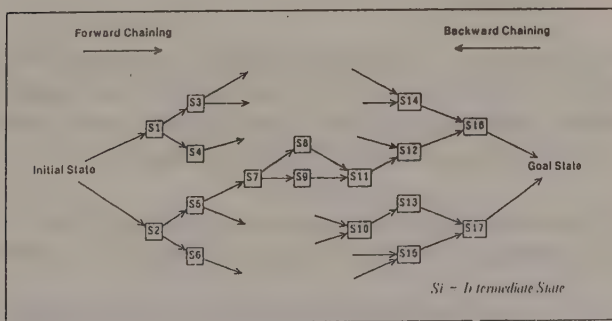


Abb. 4-5; Vorwärts- und Rückwärtsverkettung
(aus BARBER82a, S. 346)

4.1.1.2.2.3 AUTOMATISIERUNG VS. UNTERSTÜTZUNG DES PROBLEMLÖSENS

Im Zuge der Rückbesinnung auf das Anwendungsfeld Büro ist die zielorientierte Handlungsbeschreibung der im afinalen Fall genannten prozeduralen Handlungsbeschreibung überlegen, da die Operatorenauswahl erst zur Laufzeit erfolgt und die Anwendbarkeit der Operatoren in der konkreten Situation durch den Einbezug der Vorbedingungen in die Anwendungsfolge auch die Behandlung *unvorhersehbarer* Probleme ermöglicht. Das erfordert aber auch den Übergang vom klassischen Problemlösungsparadigma im Sinne der im vorhergehenden Abschnitt beschriebenen Vorwärts- und Rückwärtsverkettung auf das Paradigma der Problemlösungsunterstützung, da

- (i) die Ausgangssituation meist nicht vollständig beschrieben ist und die parallel zur systemseitigen Tätigkeit ablaufende zwischenmenschliche Kommunikation oft zur Ergänzung der Entscheidungsbasis herangezogen werden muß,
- (ii) die rein interpolative Problemlösung in eine Sackgasse führen kann und so als synthetische Komponente die Erweiterung des Operatorenvorrats (durch benutzerseitige Programmierung) erforderlich sein kann.

Den Ablauf der Problemlösungsunterstützung zeigt Abb. 4-6; weitere Ausführungen dazu finden sich in BARBER82a (S. 346ff).

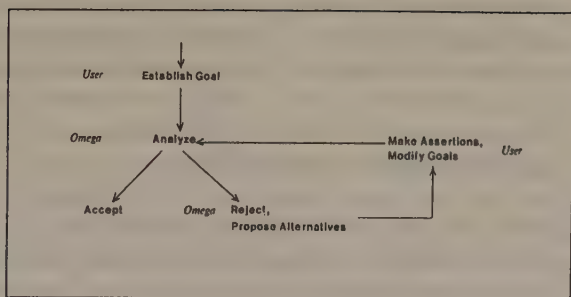


Abb. 4-6: Ablauf der Problemlösungsunterstützung
(aus BARBER82a, S. 347)

4.2 FORMULIERUNG VON BÜROPROZEDUREN

Dieser Abschnitt gibt einen Überblick über die diversen Formalismen, die zur Formulierung von Büroprozeduren eingeführt wurden. Dabei sollen auch die grundsätzlichen Aspekte, namentlich die der Motivation für die Formulierung, erörtert werden.

4.2.1 KLASSIFIKATION DER FORMULIERUNGSMECHANISMEN

Für die Formulierung von Büroprozeduren sind folgende Beweggründe ausschlaggebend:

- *Dokumentation der Organisationsstruktur und Arbeitsbeschreibungen:* Derartige Formulierungen halten die Zuständigkeiten und Autorisierungen sowie die Einzelarbeitsschritte für jeden Bürovorgang fest und können so als Instrument zur Überwachung eines geregelten Ablaufs der Bürotätigkeit dienen.
- *Durchsatz- und Durchlaufzeitabschätzungen:* In einem weiteren Schritt können nach einer zeitlichen Bewertung von Arbeitsschritten und der Festlegung von Abarbeitungs- bzw. Priorisierungsstrategien der Durchsatz bzw. Mittelwert und Grenzwerte für die Durchlaufzeit bestimmter Büroabläufe ermittelt werden. Dies dient in erster Linie der Kapazitätsplanung im Hinblick auf die Besetzung von Stellen und Dimensionierung von Kommunikationswegen.
- *Aktive Problemlösungsdienste bzw. -unterstützung:* Endziel ist dabei die Integration eines entsprechenden Steuerungsprogramms, das abhängig vom Zustand des Gesamtsystems, des betroffenen Objekts und der Eingaben entscheidungsbefugter Benutzer die Verteilung von Aufgaben in einem integrierten Büroinformationssystem steuert.

Die in diesem Bereich bekannten Modelle lassen sich analog zu der einschlägigen Unterscheidung bei Handlungsvorschriften in zwei Kategorien einteilen (vgl. hierzu auch LOCHOVSKY83b, S. 45ff) (*):

- *Prozedurale (präskriptive) Modelle:* Dieser Ansatz strukturiert Bürovorgänge anhand der Ein- und Ausgaben sowie der verarbeitenden Programme. Er ist vorwiegend für den Bürotyp *Linienbüro* geeignet und wird oft auch

(*) Eine verfeinerte Einteilung bieten Brüer et al. in ihrer Übersicht über die im Buromodellierungsbereich aktuelle Literatur an (BRÜER84 in Anlehnung an NEWMAN80). Sie unterscheiden dabei

- *Prozedurale Modelle,*
- *Informationsflußmodelle,*
- *Datenbankorientierte Modelle,*
- *Entscheidungsbasierte Modelle,*
- *Verantwortlichkeitsmodelle und*
- *Soziologische Modelle.*

Diese filigrane Unterscheidung erfaßt jedoch nicht das Grundsätzliche der Modellierungsansätze, sondern greift aus konkreten Modellen nur jeweils Aspekte heraus. Sie wird im weiteren keine Beachtung finden.

Analog dazu unterscheidet EFFELSBURG85 (S. 193)

- *Datenmodelle,*
- *Prozeßmodelle,*
- *Aktorenmodelle und*
- *Objektmodelle* (letztere werden als Kombination von Datenmodellen und Aktorenmodellen verstanden).

Auch diese Einteilung wird infolge ihrer starken Gewichtung von Implementierungstechniken hier keinen Eingang finden. Objektorientierte Modelle sind jedoch Gegenstand weiterführender Betrachtungen.

fälschlicherweise der datenbankorientierte Ansatz genannt.

- Zielorientierte (deskriptive) Modelle: Dabei wird die Büroarbeit als reine Problemlösungstätigkeit aufgefaßt. Strukturierungseinheiten sind Ziele und Strategien. Natürlich - dies sei an dieser Stelle angemerkt - mündet diese Problemlösungstätigkeit letztendlich auch in die Ausführung von Prozeduren oder Prozedurfragmenten (im Prinzip entsprechend den Basiswerkzeugen, die in Abschnitt 2.2 behandelt wurden), die als Handlungsanweisungen für das System oder den Benutzer aufzufassen sind. Der prinzipielle Unterschied zu den prozeduralen Modellen ist jedoch die der Prozedurausführung übergeordnete Kontroll- oder Steuerinstanz, die die im rein prozeduralen Ansatz vorzusehenden Kontrollstrukturen herkömmlicher Art ersetzt.

In HEWITT84 wird dieser Gegensatz durch die Dualität zwischen algorithmischer vs. problemlösender Programmierung angeführt. Hewitt und de Jong sehen dabei einen schrittweisen Abbau der für den Algorithmus geforderten Eigenschaften,

- Determiniertheit,
- Vollständigkeit und der
- Terminierungsbedingung,

um so mehr, je weniger Informationen über das Umfeld vorliegen. Schließlich wird im Zustand des reinen Problemlösens die Handlungsvorschrift zwar in einem beliebigen Umfeld einsatzfähig, in bezug auf Vollständigkeit und Terminierung können jedoch dann keine Aussagen abgegeben werden; die Determiniertheit ist in diesem Falle sekundär, da, lax ausgedrückt, im Zweifelsfall der Zweck die Mittel heiligt.

Im folgenden werden die wichtigsten Modelle auf diesem Gebiet anhand ihrer charakteristischen Eigenschaften erfaßt und gegenübergestellt. Dabei werden jedoch die Systeme nicht berücksichtigt, die unter der Flagge *Büroinformationssystem* primitive Integrationsmöglichkeiten ansonsten heterogener Werkzeuge erlauben und damit, abgesehen von einer Ausnahme, den kommerziellen Büromarkt beherrschen. Über diese wird in Abschnitt 5.1 berichtet werden.

4.2.2 PROZEDURALE MODELLE

Im Bereich der prozeduralen Modelle haben sich im wesentlichen zwei Darstellungsformen für Bürovorgänge herausgebildet, die

- *formularbearbeitenden Modelle*, denen als Basiskonstrukt das aus der Organisationstechnik bekannte Formular oder die aus der kommerziellen Datenverarbeitung bekannte Maske zugrundeliegt, und die
- *netzwerkorientierten Modelle*, die verschiedene Abarten von Petri-Netzen oder verwandte Techniken benutzen.

Formularmodelle. Formulare sind Dokumente (hier im Sinne von Abschnitt 2.3), die aufgrund spezieller Attributierung bestimmte Rahmen zur Ausgabe und Eingabe von Information bereitstellen. Die dabei zugrundeliegende Idee ist die des *Visual Programming*, d.h., ein zu bearbeitendes Büroobjekt ist *nicht* als Abstraktum definiert, sondern gemäß den in Abschnitt 3.1.1 formulierten Prinzipien direkt manipulierbar und assoziiert Operatoren an Teile der Datenobjekte vom Typ *Formular*. Im einfachen Fall kann dadurch eine eingeschränkte Programmierung des Systems durch den Endbenutzer erfolgen (vgl. dazu LOCHOVSKY83a, S. 10f oder HOFMANN84, S. 28ff).

Formulare können also sowohl, wie oben erwähnt, beim Entwurf des Büroinformationssystems als atomare Einheiten dienen als auch die Benutzerschnittstelle durch die Wiedergabe des Formulars als Bildschirmmaske beeinflussen. Ein Formular ist dabei Träger der Aktionskontrolle und des Aktionsstatus und wird in dieser Eigenschaft manipuliert, abgespeichert (in herkömmlichen Dateisätzen oder Datenbankobjekten) und insbesondere durch ein Kommunikationssystem weitergeleitet, wobei diese Weiterleitung auch durch spezielle Eigenschaften des Adressierungsschemas unterstützt werden kann (vgl. dazu Abschnitt 2.2.4).

Typische Vertreter dieser Kategorie sind im einzelnen:

- *OFFICETALK*: Die Entwicklung von *OFFICETALK* (vgl. u.a. ELLIS80a und HOFMANN84, S. 32ff) begann 1976 durch die Office Research Group am Xerox Palo Alto Research Center. Besondere Bedeutung wurde beim Entwurf der Ausgestaltung der Benutzerschnittstelle zugemessen, die sich an den Konzepten orientierte, die im Rahmen des Smalltalk-Projektes entwickelt wurden (vgl. u.a. GOLDBERG79, GOLDBERG83 und TESLER81). Die Prototypversion *OFFICETALK-ZERO* war auf eine Konfiguration von Xerox Alto 16-Bit-Minirechnern abgestimmt, die über ein lokales Netzwerk untereinander und mit einem Datenverwaltungsknoten verbunden waren. Der Ablauf von Prozeduren wurde nur unterstützt durch ein komfortables Kommunikationssystem, eine eigentliche Prozedurkontrolle wurde erst mit dem Folgesystem *OFFICETALK-D* (siehe unten) verwirklicht. Dennoch hat das System als Vorläufer des kommerziell angebotenen Informationssystems Xerox 8010 wegweisende Bedeutung.

- OBE: Als Erweiterung der Datenbank-Zugriffsmethode *Query-by-Example* (vgl. ZLOOF75) wurde für den Bürobereich am Thomas J. Watson Research Center der IBM die Sprache OBE (*Office-by-Example*) entwickelt (vgl. u.a. ZLOOF82). Der Ansatz erlaubt die Manipulation zweidimensionaler Objekte wie Tabellen, Formularen und Datenbankhierarchien sowie deren Versenden an einen vorgegebenen Empfängerkreis. Für die Programmierung von Abläufen ist die *Triggerkomponente* bedeutsam, die beim Eintritt bestimmter Ereignisse oder periodisch bestimmte Aktionsfolgen ausführt, Objekte anzeigt, sie versendet oder verändert. Durch entsprechende Steuerungsvariablen kann so auch ein iteratives Aufrufen von Triggern erreicht werden, wodurch Verzweigungen und Schleifen realisiert werden können.
- OPAS: Streng auf dem formularorientierten Ansatz beruht OPAS (*Office Procedure Automation System*) (vgl. LUM82), das am IBM Research Lab in San Jose entwickelt wurde. Dabei werden Formulare typisiert und, abhängig vom Bearbeitungszustand und den Zugriffsrechten des Bedieners, ggf. mit verschiedenen externen Repräsentationen versehen. Die Datenverwaltung ist auf einem hierarchischen Datenmodell aufgebaut. Arbeitsschritte im Büro sind als Formulartransformationen anzusehen, d.h., ein oder mehrere Formulare werden in ein Ergebnisformular transformiert. Da die für eine Transformation erforderlichen Vorbedingungen explizit formuliert werden können, ist so eine eingeschränkte Programmierung von Abläufen möglich.
- OFS: Als universitäre Entwicklung ist OFS (*Office Form System*) (vgl. TSICHRITZIS80 und TSICHRITZIS82) an der Universität von Toronto für den Betrieb in einem Netz von Kleinrechnern mit einem Standardbetriebssystem (UNIX) entworfen und implementiert worden. Jede Station

unterhält ihre eigene Datenbasis unter einem relationalen Datenbanksystem. Ansonsten ist dieses System in hohem Maße OBE nachempfunden, so daß auf eine detailliertere Beschreibung hier verzichtet wird.

Netzmodelle. Netzmodelle, dabei in erster Linie adaptierte Formen von Petri-Netzen (vgl. u.a. HERZOG84), sind allgemein als die beste und übersichtlichste Form der Darstellung von Bürovorgängen einschließlich der Nebenläufigkeit von Aktivitäten anerkannt. Der Einsatz eines Netzmodells schließt die Verwendung von Formularen als Manipulationseinheit der Bürotätigkeit nicht aus, eigentlich ist hier eher eine Ergänzung zu sehen. Jedoch legen die im folgenden beschriebenen Systeme besonderes Gewicht auf die Zusammenfassung aller Vorgangsinformationen an einer zentralen Stelle, im Gegensatz zu der eher verteilten Kontrollinformation bei formularbearbeitenden Systemen. Dabei dienen Netzmodelle jedoch nicht ausschließlich der Laufzeiterstützung, in einigen Fällen werden auch, wie oben erwähnt, quantitative Abschätzungen bzw. Simulationen auf derartigen Modellen aufgebaut. Im einzelnen sind folgende Systeme erwähnenswert

- ICN: Das Konzept der ICNs (*Information Control Net*) (vgl. u.a. ELLIS80a, S. 42ff bzw. HOFMANN84, S. 45ff) wurde im Zuge der OFFICETALK-ZERO-Entwicklung bei Xerox PARC entworfen und dient zur Beschreibung und Analyse des Kommunikations- und Informationsflusses im Büro. Zur Anwendung kam es mit der Implementierung des datenbankorientierten Folgesystems OFFICETALK-D, (vgl. BERNAL82 und ELLIS82b), das eine Beobachtungsinstanz zur Ablaufverfolgung vorsah. Dennoch konzentrierten sich die Aktivitäten im Zusammenhang mit ICN in erster Linie auf den Dokumentationsaspekt sowie die Aufdeckung von Verklebungen und Optimierung der Büroorganisation durch Simulation (vgl. ELLIS80b).

Basiselemente eines ICN sind

- das Operatorinventar,
- endliche Mengen von Quellpunkten und Endlagern,
- die Abbildung der Operatoren auf mögliche Vorgänger- und Folgeoperatoren,
- die Übergänge von Aktivitäten auf Quellpunkte und Endlager.

Abb. 4-7 zeigt ein Beispiel für die Auftragsverwaltung, wobei die durchgezogenen Linien die Übergänge von einer Operatoranwendung zur nächsten und die gestrichelten Linien den Datenfluß von den Quellpunkten und zu den Endlagern symbolisieren.

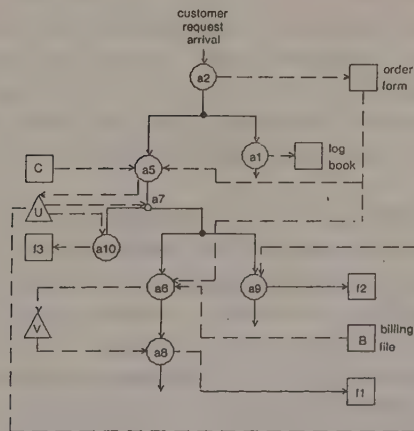


Abb. 4-7: Beispiel für ein Information Control Net
(aus ELLIS80a, S. 45)

Als weitere Verfeinerung des ICN wurde im Jahre 1983 das GICN (*Generalized Information Control Net*) (vgl. auch ELLIS83) vorgestellt, das zusätzlich noch Zeitbedingungen und Fertigstellungstermine aufnehmen konnte.

- SCOOP: Im Jahre 1977 vorgestellt, ist das an der Universität von Pennsylvania von Michael Zisman entwickelte SCOOP (*System for Computerization of Office Processing*) (vgl. ZISMAN77 und ELLIS80a, S. 33ff) einer der frühesten Versuche, Spezifikation, Repräsentation und Automatisierung von Büroprozeduren einem einheitlichen Gesamtkonzept zu unterstellen. Der Ansatz beruht auf dem Konzept der *Petri-Netze* und assoziiert an die Transitionen Produktionsregeln, d.h. Paare aus Prämissen und daraufhin auszulösenden Aktionen. Da bei der Abarbeitung immer nur die Prämissen überwacht werden müssen, deren assoziierte Transitionen zur Auslösung anstehen, findet so eine sinnvolle Partitionierung der durch die Produktionen konstituierten Wissensbasis statt. Derartige Beschreibungen werden für jede der im Büro vorzusehenden Instanzen angefertigt, wodurch auch eine individuelle Gestaltung der Kommunikationsprotokolle möglich wird.
- PrT-Netze: Als Erweiterung der auf der Petri-Technik beruhenden *Kanal/Instanz-Netze* (vgl. RICHTER83, S. 212ff) wurden für den Büroeinsatz auch die sog. *Prädikat/Transitionenetze* (PrT-Netze) vorgeschlagen (vgl. RICHTER83, S.28ff). Dabei werden die Stellen eines Petri-Netzes als Prädikate aufgefaßt, die den an dieser Stelle befindlichen Objekten (anstelle von Markierungen) zugesprochen oder abgesprochen werden. Den Transitionen ist, ähnlich wie in SCOOP eine Transitionsregel assoziiert, die Trans-

formation, Vernichtung und Neuschaffung von Objekten steuert. Ein Beispiel (Darstellung des Arbeitsablaufes in einer Arbeitsvermittlungsstelle) zeigt Abb. 4-8.

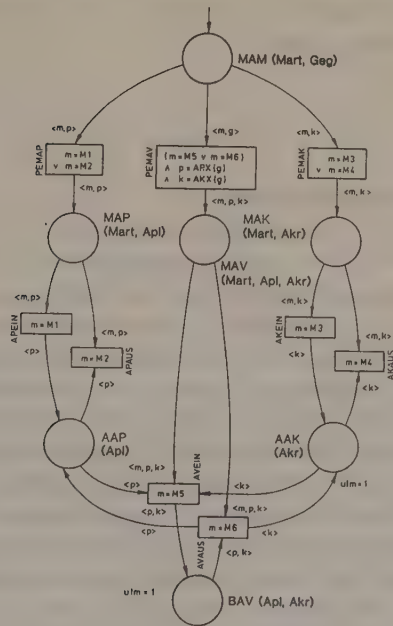


Abb. 4-8: Beispiel der Ablaufdarstellung mit
PrT-Netzen (aus RICHTER84, S. 32)

Auf der Basis dieses Modells wurde u.a. ein Simulator entwickelt, der das Problem der Erreichbarkeit direkt behandelt: Er löst die Frage, mit welcher Schaltfolge aus

einer gegebenen Anfangsbesetzung eine Endbesetzung mit bestimmten vorgegebenen Eigenschaften erreicht werden kann (vgl. NIEHUIS85).

Bereits an der Auswahl der Beispiele wird sichtbar, daß hier nur reine Routinevorgänge wiedergegeben werden, die explizit als Aktionsabfolge bereitstehen.

4.2.3 ZIELORIENTIERTE MODELLE

Zielorientierte Modelle beruhen auf einer ablaufunabhängigen Beschreibung des Büros sowie der Formulierung von Problemen im Sinne der in Abschnitt 4.1.2.1 vorgestellten afinalen Handlungsvorschriften. Grundlegend hierfür war die Feststellung (so z.B. in FIKES80 und SUCHMAN83), daß der Handlungsraum in der Büroumgebung in praxi nicht abgeschlossen ist und deshalb die ad-hoc-Erstellung von Büroprozeduren erforderlich ist. Die Arbeiten auf diesem Gebiet sind in hohem Maße von den parallelen Arbeiten und der Begriffswelt im Gebiet der Künstlichen Intelligenz geprägt. Der Büroarbeit können dabei (nach BARBER82a, S. 343f) vier Charakteristika zugesprochen werden (vgl. Abb 4-9):

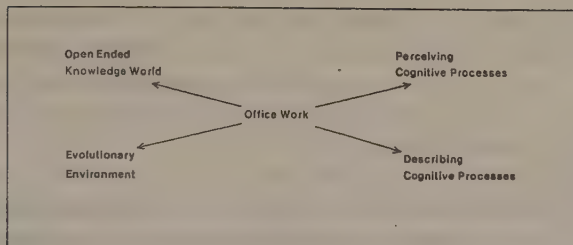


Abb. 4-9: Charakteristika der Büroarbeit nach dem Problemlösungsansatz (aus BARBER82a, S. 343)

- **Nichtabgeschlossene Wissenswelt:** Im Gegensatz zu den kanonischen Welten der Künstlichen Intelligenz, so z.B. der *Blockwelt* oder der *Genealogie* mit den berühmten Großvaterbeispielen, ist die Wissenswelt des Büros nicht abgeschlossen: Die Menge aller möglichen Zustände ist nicht ermittelbar, gleichsam alle möglichen Alternativen zur Erreichung eines Ziels. Dies untermauert nochmals die Forderung nach einem System, das unvorhersehbare Situation bewältigen kann (*).
- **Evolutionäre Umgebung:** Organisationen sind ständig im Wandel. Ein klassisches Problem des Software Engineering ist die Zeit zwischen der Ist-Analyse und Anforderungsspezifikation und der Einführung des ausgetesteten Programmprodukts: Entweder ist in diesem Zeitraum die Organisation "eingefroren" worden, oder es tritt bereits zum Einführungszeitpunkt eine Diskrepanz zwischen Anwendungsprogramm und der korrespondierenden

Organisationsstruktur auf, so daß an diesem Punkt bereits erste Revisionen vonnöten sind. In der Büroorganisation ist es erforderlich, die Organisationsstruktur als zeitliche Folge verschiedener Versionen darzustellen, um die Nachvollziehbarkeit auf der Basis einer bereits überholten Struktur getroffener Entscheidungen sicherzustellen.

- *Rückschlüsse von Vorgängen auf kognitive Prozesse:* Selbst bei einfacheren kognitiven Prozessen kann für deren Beschreibung die Vollständigkeit der einflußnehmenden Regeln und Randbedingungen nicht sichergestellt werden, insbesondere dann nicht, wenn eine derartige Beschreibung auf Rückschlüssen von Vorgängen und Benutzerreaktionen beruht. Dies ist ein weiteres Argument für die Rücknahme des Ziels der Problemlösungsautomation auf das der Problemlösungsunterstützung.
- *Beschreibung mentaler Prozesse:* Die Beschreibung mentaler Prozesse, die bei der Simulation des Problemlösungsvorgangs herangezogen wird, sollte nicht auf der Vorstellung eines Standardbenutzers basieren,

(*) Selbstverständlich bezieht sich diese Bemerkung auf die real auftretenden Phänomene der Bürowelt. Nach der Abbildung dieser Phänomene in die Datenstrukturen eines Büromodells auf einem Automationssystem ist selbstverständlich die Endlichkeit gegeben, da ein derartiger Automat nur abzählbar-endlich viele Zustände einnehmen kann. Ist jedoch auch die Adäquatheit dieses Abstraktionsprozesses zur Diskussion gestellt, wird also unterstellt, daß der Informationsverlust bei dieser Abstraktion zu inadäquaten Systemreaktionen führt, ist durchaus die Nichtabgeschlossenheit zu konstatieren.

sondern durchaus die individuellen Eigenarten der Beteiligten mit in Betracht ziehen. Denkbar wäre hier eine erweiterte Benutzermodellierung nach dem Vorbild der in Abschnitt 3.1.1 dargestellten Benutzermodelle in der Dialoggestaltung.

Im folgenden werden einzelne Aspekte der zielorientierten Büromodellierung beleuchtet, wobei besonders auf den Einsatz von Datenbankkonzepten zur Problemlösung Bezug genommen wird.

Bürowissen - Struktur der zugrundeliegenden Wissensbasis. Die Basis des zielgetriebenen Problemlösungsansatzes im Büro bildet die sog. *Bürosemantik* (vgl. BARBER82b, S. 364ff und BARBER83, S. 46), die versucht, der Intention hinter den im Büro zu beobachtenden Phänomenen auf die Spur zu kommen. Betrachtet werden dabei die *Organisationsstruktur* einerseits, das *Anwendungswissen* andererseits (vgl. Abb. 4-10).

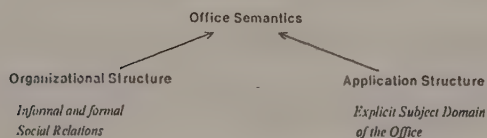


Abb. 4-10: Konstitution des Begriffs Bürosemantik
(aus BARBER83, S. 46)

Die Organisationsstruktur beschreibt die formellen und informellen Beziehungen der an der Organisation Beteiligten. Dabei sind selbstverständlich Vorgesetzten/Untergebenen-Beziehungen zu

beachten, aber auch individuelle Zuständigkeiten, Kompetenzen und Fähigkeiten. Die Applikationsstruktur betrifft den eigentlichen und offensichtlichen Zweck des Büros.

Die meisten problemlösenden Experimentalsysteme sind von Expertensystemen abgeleitet und stellen den Bezug zu der Büroumgebung nur durch die Wahl des Anwendungsfeldes her. Zu nennen ist dabei wohl insbesondere ODYSSEY (vgl. FIKES81), das für die Reiseplanung (Routen- und Hotelauswahl) Unterstützung bietet. Dennoch werden derartige Systeme der obengenannten Forderungen nicht gerecht und sind eher als spezialisierte Decision-Support-Systeme anzusehen.

Eine Prototypentwicklung eines bürobezogenen problemlösenden Systems ist OMEGA, das am Massachusetts Institute of Technology entwickelt wurde. OMEGA benutzt Objektbeschreibungen, die in ihrer Externen Repräsentation als Formulare sichtbar werden. Folgende Funktionen werden von herkömmlichen Formularen auf OMEGA-Beschreibungen übernommen (vgl. BARBER83, S. 51):

- Ablage von Information wie in Datensätzen,
- Übertragung von Information wie durch Nachrichten,
- Darstellung von Information,
- Aggregation und Änderung von Information.

Darüber hinaus jedoch enthalten sie:

- Fehlerentdeckung und Plausibilitätsüberwachung,
- Deduktive Komponenten zur Ableitung von Information aus der Datenbasis,

- Strukturbeschreibungen für Bürotätigkeit,
- Strukturbeschreibungen für die Organisation und die Anwendungswelt,
- Standpunkte (viewpoints), die das Argumentieren über inkonsistente Information erlauben, namentlich bei einer Änderung der vorgenannten Strukturbeschreibungen.

Standpunkte sind Sammlungen von Beschreibungen für Objekte. Änderungen in der Anwendungswelt werden nicht durch eine Änderung einer Beschreibung wiedergegeben, sondern als neuer Standpunkt in das System eingebracht. Im Detail ist der Ablauf bei der Anlage eines neuen Standpunkts wie folgt (sinngemäß nach BARBER83, S. 52):

1. Ein neuer Standpunkt wird angelegt und als Nachfolger des alten Standpunktes gekennzeichnet.
2. Alle Beschreibungen, die nicht von der Änderung abzuleiten sind, werden vom alten Standpunkt vererbt.
3. Die Änderung wird in den neuen Standpunkt eingetragen.
4. Alle von den Änderungen abhängigen Beschreibungen werden in den neuen Standpunkt eingetragen.

Standpunkte sind also mit dem bislang intuitiv eingeführten Begriff Kontext gleichzusetzen. Bei Kontextänderungen müssen Prozeduren zur Verfügung stehen, die den korrekten Zustandsübergang überwachen, bzw., die ggf. den Übergang als illegal erklären und verhindern. Die eigentliche Problemlösung erfolgt im Backward-Chaining-Verfahren durch Auflösung von Zielen in Unterziele, wobei im Sinne der Problemlösungsunterstützung die in Abb. 4-6 dargestellte Vorgehensweise gewählt wird. Besondere Beachtung verdient das angewendete Konfliktlösungsverfahren, das

bei Widersprüchen von dem betroffenen Standpunkt Zustandsrechtfertigungen anfordert und nach deren Abgleich einen neuen Standpunkt mit der abgeglichenen Information bildet.

Ausnahmesituationen als Probleme - der datenbankorientierte Problemlösungsansatz. Neben den betrachteten Modellen, die sich in Infrastruktur und Lösungsweg an bereits bekannten Techniken der Künstlichen Intelligenz orientieren, sind auch Ansätze möglich, die Datenbanktechniken problemlösend einsetzen. Da Probleme im Zusammenhang mit Datenbanksystemen nicht *explizit* formuliert werden können, muß durch die Einführung eines Problembegriffs, der sich an datenbanktechnischen Stereotypen orientiert, die Behandlung von Problemen angegangen werden.

Für die von Datenbanksystemen verwalteten Objekte können im Lauf ihrer Lebenszeit zwei Phasen unterschieden werden (vgl. auch KRATZER85):

- die *dynamische Phase*, oder auch *Aufbauphase*, die die Entstehung und die, entsprechend den Querbeziehungen zu anderen Objekten, nachträgliche Modifikation von Datenobjekten umfaßt, und
- die *statische Phase*, oder auch *Archivphase*, in der das Objekt unverändert für rein historische oder statistische Zwecke bereitsteht.

In der dynamischen Phase, die sich im Brennpunkt der weiteren Betrachtungen befindet, können zweierlei Konflikte eintreten, die die weitere Verwendung eines Objektes einschränken:

- (i) *Unvollständigkeit:* Durch unvollständige Besetzung der Objektbeschreibung kann der Einsatz des Objektes eingeschränkt sein; dies kann entweder dadurch eintreten, daß ein Beschreibungsattribut bei einer Abfrage auf das Objekt nicht ausgewertet werden kann, oder daß die

Beziehungen zwischen Objekten aufgrund fehlender Beschreibungsattribute nicht aufgebaut bzw. überwacht werden können.

- (ii) *Integritätskonflikt*: Der Zustand einer Objektwelt wird durch Integritätsbedingungen bezüglich seiner Integrität im Hinblick auf die internen datenmodellabhängigen Konsistenzanforderungen beschrieben, aber auch im Hinblick auf die Adäquatheit zu den Gegebenheiten der Anwendungswelt. Im ersten Fall spricht man von *Konstitutiven Integritätsbedingungen*, im letzteren von *Regulativen Integritätsbedingungen* (vgl. WEDEKIND84). Bei einem Konflikt zwischen den Integritätsbedingungen und dem Zustand der Objektwelt gibt es folgende Vorgehensweisen der Integritätsüberwachung und -erhaltung:

- Die *Präventive Strategie* verhindert Integritätskonflikte a priori durch Ablehnung von Operationen, die die Objektwelt in einen inkonsistenten Zustand versetzen.
- Die *Permissive Strategie* läßt alle Änderungen der Objektwelt prinzipiell zu, gibt aber auf Anfrage Kritikpunkte am Zustand, d.h. eine Sammlung der durch den Zustand der Objektwelt verletzten Integritätsbedingungen, aus.
- Die *Reparative Strategie* läßt ebenfalls alle Änderungen der Objektwelt zu, versucht jedoch eigenständig eine Anpassung der vom Nachzustand einer Manipulationsoperation erfaßten Integritätssphäre und geht erst bei Mißerfolg auf die Präventive Strategie über.

Jede dieser beiden Anomalien kann als *Ausnahmesituation* aufgefaßt werden. Darauf aufbauend kann der Problembegriff in der

Datenbankwelt folgendermaßen formuliert werden:

(i) Die Unvollständigkeit von Objekten wird in der Datenbanktechnik auf Nullwerte der Attribute von Datenbankobjekten abgebildet. Nullwerte an sich sind Träger keines konkreten Wertes, können jedoch durch logische wie modallogische Aussagen weitergehend präzisiert sein. Modallogisch geschieht das durch Konstatierung der Möglichkeit bzw. Unmöglichkeit der Festlegung eines konkreten Wertes anstelle eines Nullwertes (*), logisch durch die Einführung zusätzlicher Bedingungen, die bei der Substitution eines Nullwertes durch einen konkreten Wert für diesen zu gelten haben (z.B. die Äquivalenz zu der konkreten Ausprägung eines anderen Nullwertes). Betrachtungswürdig sind nur Nullwerte, die potentiell durch einen konkreten Wert ersetzt werden können. Probleme werden deshalb durch Objekte konstituiert, die derartige Nullwerte enthalten; das Ersetzen dieser Nullwerte durch konkrete Werte entspricht der Problemlösung. Die direkte Analogie zu der Tabellenverwaltung mit Kalkulationsprogrammen, wobei die Nullwerte der Tabelle oder des Formulars den abhängigen Zellen entsprechen (vgl. Abschnitt 2.2.2), ist offensichtlich, nur ist hier der Lösungsweg deterministisch, also quasi-prozedural, vorgegeben.

(ii) Schwieriger gestaltet sich das Problem bei Integritätskonflikten. Für die Erschließung des Problembegriffs aus dieser Art von Phänomenen ist einerseits die *betroffene Integritätsbedingung* zu beachten, andererseits aber auch das *auslösende Daten-*

(*) WEDEKIND83 greift diesen Gedanken auf und fordert eine *Modallogische Normalform*, die die Anomalie der Unmöglichkeit eliminiert.

objekt sowie der auslösende Operator. Die Vorgehensweise beruht auf der Annahme, daß das Paar (Datenobjekt, Operator) in jedem Falle Teil der anzustrebenden Zielsituation ist, wobei jedoch infolge der Integritätsbedingungen weitere Änderungen der Objektwelt erforderlich sind, um die Zielsituation auch in integrem Zustand zu erreichen. Probleme dieser Art sind also als *partiell-erreichte Zielsituationen* aufzufassen, die im Zuge der Problemlösung in *konsistent-erreichte Zielsituationen* übergeführt werden müssen. Dabei muß das auslösende Paar (Datenobjekt, Operator) entweder erhalten bleiben oder kontrolliert durch entsprechende Paare substituiert werden.

In beiden Fällen zeigt sich wiederum die Äquivalenz zwischen Problemlösen einerseits und dem Umsetzen der theoretischen Modalität *möglich* in die praktische Modalität *erreichbar* andererseits. Wie ist nun die technische Realisierung eines derartigen Umsetzungsvorgangs denkbar? Der naheliegende Gedanke ist die Einführung von Prozeduren (d.h. Operatoren), die an Ereignisse gekoppelt sind und durch diese ausgelöst werden. Diese Art von Prozeduren wird als *Trigger* bezeichnet; die zahllosen Veröffentlichungen zu diesem Thema basieren auf der Präsentation des Triggerkonzeptes im Rahmen der Entwicklung eines der ersten funktionsfähigen relationalen Datenbanksysteme, System R, im IBM Forschungslabor San Jose (vgl. ESWARAN76a). Darauf aufbauend schlägt KRATZER85 einen Mechanismus vor, der Prozeduren in der Metadatenbank des Datenbanksystems an Objektbeschreibungen für die Behandlung von Nullwerten und an Integritätsbedingungen für die Auflösung von Integritätskonflikten bindet. Der hierfür gepragte Fachterminus ist *Legende*, der die Zuordnung eines Prozedurkomplexes an ein Attribut oder eine Integritätsbedingung bezeichnet. Das Legendenkonzept enthält eine implizit vorgegebene Kontrollstruktur, die aus den innerhalb einer Legende zusammengefaßten Prozedurfragmenten das relevante auswählt, wobei folgende Kriterien die Auswahl bestimmen:

- Art des auslösenden Ereignisses,
- Art des auslösenden Operators (*),
- Tageszeit und Datum,
- Benutzeridentifikation,
- Ausweichoptionen im Sinne von ggf. bei Mißerfolg anzuwendenden Alternativoperatoren,
- Auswertungsoptionen (im Fall der Nullwertlegende entweder bei Eintrag oder bei lesendem Zugriff).

Die Einbettung der Legenden in die Metadatenbank erlaubt auch die Ausnutzung der dort vorhandenen Strukturen, die der einfachen Objektbeschreibung übergeordnet sind: So ist auch eine Legendendefinition auf Rollennamen- oder Wertebereichebene vorstellbar, die im Bedarfsfall auf der tieferen Ebene der Einzelattributbeschreibung präzisiert werden kann.

Da im Zuge der Abarbeitung wiederum neue Legenden ausgelöst werden können, ist damit eine Abarbeitungsstrategie realisierbar, die analog zum *Backward Chaining* als Verarbeitungsrichtung abläuft. Dies ist jeweils der Fall, wenn innerhalb eines aktiven Prozedurfragmentes wiederum eine Problemstellung im oben genannten Sinn als Zwischenziel erzeugt wird.

(*) Dabei ist nicht nur der Operator selbst gemeint, wie beispielsweise bei ESWARAN76a (S. 11) die Operatorbezeichnung auf der Ebene der primitiven Datenbankoperationen (INSERT, UPDATE, DELETE), sondern auch der Gesamtkontext der Operation, d.h. insbesondere die Qualitätsanforderungen der übergeordneten Anwendung.

Eine Pilotimplementierung dieses Konzepts in einer kommerziellen Datenbankumgebung ist in FRIEDRICH86 und KASBERGER86 beschrieben; dabei wurden sowohl ein Laufzeitsystem als auch die Einbettung in die Verwaltung der Metadatenbank realisiert.

Weitergehende Betrachtungen zu Sprachkonstrukten zur Behandlung von Ausnahmesituationen finden sich insbesondere in BORGIDA85; dabei werden u.a. auch Typverletzungen einbezogen, die jedoch keinen direkten Bezug zur Büroumgebung haben.

Anmerkungen zu objektorientierten Systemen als Basismodell.
Dieser Teilabschnitt ist dem Basismodell gewidmet, das eindeutig die Diskussion beherrscht, die über sowohl für Dokumente und Visualisierung als auch für die Modellierung von Büroarbeit geeigneten Datenmodelle geführt wird. Dieses Daten- und Ausführungsmodell ist im Grunde nur eine Stilrichtung der Strukturierung von Softwaresystemen (vgl. hierzu STOYAN84) und wird aufgrund der engen Kopplung zwischen Datenstrukturen und zugehörigen Prozedurfragmenten *Objektorientierte Programmierung* genannt.

Was sind die Anforderungen an eine Objektorientierte Umgebung? STOYAN83 nennt als Grundbegriffe den *Nachrichtenaustausch* und die *Hierarchisierung* und des weiteren folgende Basiselemente:

- *Objekte:* In einem Objektorientierten System ist jeder Gegenstand ein Objekt, das über einen eindeutigen Namen erreicht werden kann bzw. durch ein intermediäres Objekt vermittelt wird. Objekte können Bezug zur Realwelt haben, indem sie das Verhalten realer Objekte simulieren, oder auch reine systemtechnische Artefakte sein. Objekte können verschiedene innere Zustände annehmen.

- *Nachrichten:* Sie dienen der Aktivierung von Objekten. Bei Empfang einer Nachricht durch ein Objekt reagiert dieses (nach STOYAN83, S. 11) gemäß einem der folgenden Abläufe:

- (i) Das Objekt antwortet direkt mit einer Quittungsnachricht.
- (ii) Das Objekt verändert seinen inneren Zustand und antwortet dann direkt mit einer Quittungsnachricht.
- (iii) Das Objekt sendet eine Nachricht an ein anderes Objekt. Dieses antwortet dem ursprünglichen entweder direkt oder durch Vermittlung des zwischengeschalteten Objekts.
- (iv) Das Objekt leitet aus der empfangenen Nachricht eine oder mehrere Sendungen an andere Objekte ab, versendet diese und quittiert anschließend, ggf. nach Veränderung seines eigenen inneren Zustands.
- (v) Das Objekt leitet aus der empfangenen Nachricht eine oder mehrere Sendungen an andere Objekte ab, versendet diese, ggf. nach Änderung seines eigenen inneren Zustands, und eines der aktivierten Objekte versendet die Quittungsnachricht an den ursprünglichen Sender.
- (vi) Das Objekt weist die Nachricht zurück.

Damit lassen sich die verfügbaren Elementarhandlungen auf folgende drei reduzieren:

- Nachrichten senden (unter Einschluß des Sendens der Quittungsnachricht),
- Nachrichten empfangen,
- den inneren Zustand ändern.

Nachrichten können zu Sequenzen zusammengefaßt werden, die entweder stückweise von den Adressaten verarbeitet werden oder nacheinander abgeschickt werden.

Zu jedem Objekt ist das von ihm verarbeitete *Protokoll* bekannt, das aus der Gesamtheit der von ihm verstandenen Nachrichten besteht. Beim Versand einer Nachricht ist deshalb zwar keine Detailinformation über die Verarbeitung der Nachricht beim Empfängerobjekt erforderlich, aber über die wechselseitig einzuhaltenden Protokolle muß ein Einverständnis herrschen. Auf dieser Basis lassen sich aus der primitiven Nachrichtenübertragung komplexere Protokolle, z.B. von simplen Prozeduraufrufprotokollen bis zu zwischenmenschlichen Kommunikationsprotokollen, simulieren. Zusammen mit dem inneren Zustand wird so die Verwandtschaft zu Abstrakten Datentypen (vgl. GUTTAG77) deutlich.

Die Verwandtschaftsbeziehungen zwischen Objekten sind durch die sog. *Klasse-Instanz-Beziehung* sowie die *Klasse-Oberklasse-Beziehung* geprägt. Objekte werden als *Instanzen* von *Klassen* gebildet, die selbst wiederum Objekte sind. Damit ist die *Klasse-Instanz-Beziehung* nach Abschluß des Erzeugungsvorgangs konstitutiv gegeben. Den Objekten, die Instanzen einer Klasse sind, ist das jeweils gültige Protokoll gemeinsam. Dabei wird das Protokoll durch die sog. *Methoden* spezifiziert, die an das Klassenobjekt assoziiert sind. Zugleich beschreibt das Klassenobjekt auch die Menge der inneren Zustände, die ein Objekt annehmen kann, d.h. die "Adressen" (*slots*) untergeordneter Objekte, die das betroffene Objekt beschreiben. Nimmt man einige grobe Vereinfachungen in Kauf, ist bei Einschränkung der Sicht

auf die Beschreibung von Datenstrukturen im objektorientierten Modell eine Analogie von Klassen zu Relationenschemata, Instanzen zu Tupeln einer Relation und Slots zu Attributen durchaus zulässig.

Die andere Beziehungsart *Klasse-Oberklasse-Beziehung* drückt Ähnlichkeiten zwischen Klassen im Sinne einer Art-Gattungs-Hierarchie aus. Hierdurch entsteht entweder, bei strikter Zugehörigkeit einer Art zu nur einer Gattung, eine Hierarchie von Klassenbeschreibungen, bei Unterlassung der Einschränkung ein gerichteter, zyklentreier Graph (*).

Zentral für den Einsatz objektorientierter Systeme ist das Konzept der *Vererbung*, das die Übernahme der Methoden und Slots von den hinsichtlich der Klasse-Oberklasse-Beziehung übergeordneten Klassen auf die jeweils untergeordneten Klassen vorsieht. Anders ausgedrückt heißt dies, daß eine Unterklasse als Spezialisierung der Oberklasse aufzufassen ist und mithin alle Zustandsbeschreibungen und auch das Protokoll zunächst übernimmt, bevor im Zuge des Spezialisierungsprozesses die Ergänzung der Beschreibungsattribute bzw. Substitution der allgemeinen Protokollregeln auf spezielle hin erfolgt.

CARDELLI⁸⁵ formuliert hinsichtlich der Typenbildung in Objektorientierten Systemen folgende *notwendige* Eigenschaften (S. 481):

- Objekte seien Datenabstraktionen mit einer Schnittstelle, die durch Operationen gebildet wird, die mit einem Namen angesprochen werden können, sowie einem inneren Zustand, der nur über diese Operatoren sichtbar wird.

(*) Die dabei auftretenden Probleme und Konfliktsituationen für den Fall der Smalltalk-80-Umgebung behandelt BORNING⁸².

- Objekte sind einem Typ bzw. einer Klasse zugeordnet.
- Typen erben Attribute von übergeordneten Typen.

Referenzsysteme, die den objektorientierten Programmierstil direkt unterstützen, sind die in der 60er Jahren entwickelte Sprache SIMULA, die Experimentalsysteme zur Aktoretheorie, die Erweiterungen der Sprache LISP um *Flavors* und insbesondere das am Xerox Palo Alto Research Center entwickelte System Smalltalk-80 (vgl. zur Geschichte der objektorientierten Systeme KOVED84, zu Smalltalk-80 insbes. GOLDBERG83 und zu Aktoren HEWITT73).

Das Prädikat *objektorientiert* wird selbstverständlich oft nur in sehr laxer Form gebraucht: In der Datenbankwelt ist oft schon eine geringfügige Anpassungsfähigkeit des Systems an die Interpretation bestimmter Attribute hinreichender Grund, von einem "objektorientierten" System zu sprechen. Ein markantes Beispiel hierfür ist DITTRICH85. In diesem Sinn kann auch der in Abschnitt 2.3.1 vorgestellte Dokumentmonitor als objektorientiertes System aufgefaßt werden. Bei Datenbanksystemen ist für eine sinnentsprechende Verwendung des Begriffs eine Abkehr von herkömmlichen Datenmodellen erforderlich, gleichermaßen die Ergänzung um ein Programmiersystem, das die eigentlichen Datenbankzugriffe verdeckt und ausschließlich auf der internen Ebene (nach der Terminologie der ANSI/SPARC-Architektur von 1975) durch Festlegung von Qualitätsstufen hinsichtlich der Datenverwaltungsdienste die physische Abspeicherung von Objekten in einer virtuell-unendlichen Objektwelt regelt. Ansätze hierfür sind u.a. in ZIESCHANG85 oder im Umfeld des Datenbanksystems INGRES in STONEBRAKER83a, STONEBRAKER83b, STONEBRAKER84, ONG84 und STONEBRAKER85 zu finden. COPELAND84 beschreibt eine Integration der objektorientierten und Datenbanktechniken auf der Basis einer Smalltalk-Umgebung. Die Mängel von Datenbanksystemen werden dabei wie folgt angeführt:

- *Ungenügende Typisierung:* Herkömmliche Datenbanksysteme verfügen nur über eingeschränkte Möglichkeiten der Typenbildung auf Attributebene. Selbst die im CODASYL-Modell vorgesehene Aggregationsmöglichkeit elementarer Datenobjekt zu Gruppen ist keine, wenn auch primitive, Typisierung, sondern nur ein organisatorisches Konstrukt zur besseren Handhabung der Datenstruktur von der Wirtssprache aus.
- *Technische Restriktionen:* Die Modellierungsfähigkeit wird durch die technischen Restriktionen, die aus dem Datenmodell und seinen zugrundeliegenden Implementierungskonzepten erwachsen bzw. aus der Wirtssprache und der Betriebssystemumgebung übernommen werden, in hohem Maße eingeschränkt.
- *Strukturelle Einschränkungen:* Herkömmliche Datenmodelle sind zur Wiedergabe der Irregularitäten der Anwendungswelt ungeeignet. Die Gründe liegen in der strengen Typisierung (Relationales Datenmodell) (*), Einschränkungen der Komplexbildung (z.B. Verbot rekursiver Datenstrukturen im CODASYL-Datenmodell) oder der Unvollständigkeit (Hierarchisches Datenmodell, z.B. im Zusammenhang mit der Abbildung von n:m-Beziehungen).
- *Unzureichende Modellierungsmöglichkeiten:* Die Abbildung realweltlicher Phänomene auf die Strukturen herkömmlicher Datenbanksysteme erfordert eine übermäßige Vereinfachung und die totale Absicherung und Überwachung der möglichen Datenbankzustände durch Integritätsbedingungen und ein entsprechendes Subsystem. Durch die Aufnahme von Prozedurfragmenten zur Manipulation von Objektkomplexen höherer Ordnung in die Datenbeschreibung kann zwar nicht auf die Integritätsbedingungen verzichtet werden, hingegen ist es möglich, weitestgehend auf Laufzeitüberwachungen zu verzichten,

da beim Datenbankentwurf bereits viele Bedingungen erkannt werden können, die durch die einzusetzenden Prozedurfragmente implizit eingehalten werden.

Copeland et al. (COPELAND84) entwickelten ein Mengenkonzept auf der Basis des Smalltalk-80-Systems, das die Abfragemöglichkeit durch mengentheoretische Konstrukte vorsah.

Der Einsatz Objektorientierter Systeme in Büroanwendungen ist in allen bisher betrachteten Bereichen bedeutsam (vgl. auch die allgemeinen Ausführungen in GLADNEY85):

- In der Dokumentenerstellung und -manipulation kann die in Abschnitt 2.3.1 eingeführte Struktur zur Dokumentverwaltung direkt in das System eingebracht werden, ohne den Umweg über eine Prozeßhierarchie nehmen zu müssen. Beispiele für derartige Systeme sind in LYNGBAEK84, HORAK84a und AHLSEN84 beschrieben.

--- -----

- (*) So ist z.B. der in FIKES80 genannte Fall, daß ein Attribut vom Typ einer ganzen Zahl durch ein Prozedurfragment zu ersetzen ist, nur durch Kunstgriffe und nicht im allgemeinen Fall auf das relationale Datenmodell abbildbar. In einer Smalltalk-80-Umgebung ist durch das Blockkonzept, d.h. die Möglichkeit, aufgrund der fehlenden Typisierung einen Wert durch eine Nachrichtenfolge mit dem entsprechenden Rückgabeobjekt zu ersetzen, die nachträgliche Ermittlung von Werten möglich. Allerdings sind in jüngster Zeit auch Typisierungsaktivitäten in Objektorientierten Systemen bekanntgeworden, die eine Einschränkung dieser Möglichkeit bedeuten können (vgl. u.a. MAUL85). Es sollte jedoch in jedem Fall ein polymorphes (CARDELLI85, S. 475) Typsystem erhalten bleiben, um die Vorzüge der Objektorientierten Programmierung gegenüber den herkömmlichen Umgebungen nicht einzubüßen.

- In der *Präsentationsverwaltung und Benutzermodellierung* zählen Objektorientierte Modelle in jedem Falle zu den tragenden Elementen der Weiterentwicklung. Beispielhaft ist hierfür die Fensterverwaltung des Smalltalk-80-Systems (vgl. TESLER81, S. 100) bzw. die entsprechende Entwicklungen zum Xerox System 8010 (siehe u.a. SMITH82).
- In der *Ablaufverwaltung für Bürohandlungen* sind bereits diverse Modelle zur Abbildung und Überwachung von Handlungen ausgearbeitet worden. Stellvertretend für die verschiedenen möglichen Konzepte ist in Abb. 4-11 ein entsprechender Architekturvorschlag dargestellt.

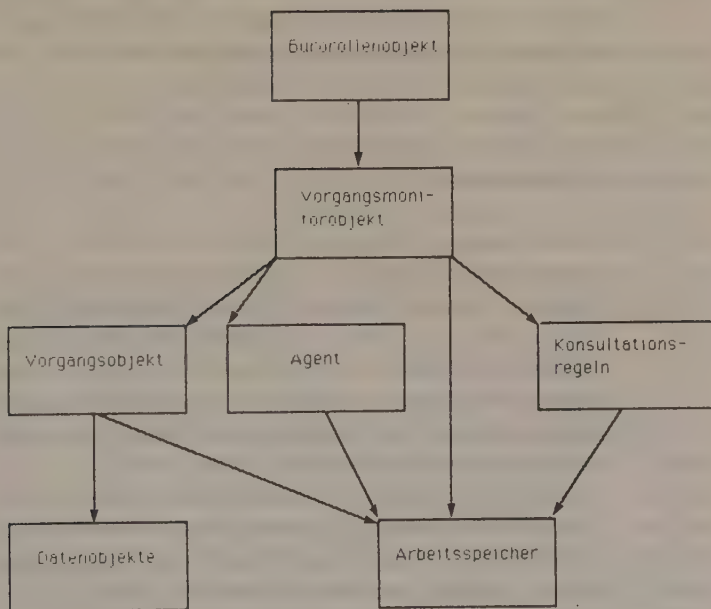


Abb. 4-11: Ablaufüberwachung in Objektorientierten Modellen (nach W0085, S. 16)

Das Modell von Woo und Lochovsky (vgl. W0085) entspricht in groben Zügen den im Smalltalk-80-System entwickelten Vorstellungen; Objekte können explizit an

andere Objekte assoziiert werden, der modifizierbare innere Zustand ist durch eine Liste von Variablen beschrieben, Methoden werden auf Regeln abgebildet, die gleichzeitig auch die Rolle von Triggern übernehmen können. Zwei ausgezeichnete Regeln, die ALPHA- und die OMEGA-Regel, sind für das Anlegen und Beseitigen von Instanzen einer Objektklasse vorgesehen.

Die verschiedenen Kategorien von Objekten können folgendermaßen beschrieben werden:

- *Bürorollenobjekte* sind für die Simulation einer bestimmten Bürofunktion verantwortlich; "Abteilungsleiter", "OP-Sachbearbeiter" oder "CAD-Konstrukteur" sind Beispiele für derartige organisatorische Stereotype. Damit sind beliebige funktionale Aufteilungen abbildbar. Über eine spezielle Ein-/Ausgaberegeln kann die Kommunikation mit den anderen Büroobjekten stattfinden.
- *Datenobjekte* sind die Basis der herkömmlichen Datenverarbeitung; die Modellierung erfolgt analog zum Datenbankdesign; durch die Zugabe komplexerer Regeln können Datenobjekte autonom als Trigger verwendet werden, z.B. um Quittungen, Erinnerungen u.ä. zu versenden.
- *Vorgangsobjekte* modellieren Büroprozeduren. Dabei können zwei Fälle unterschieden werden: Falls eine *afinale*, d.h. prozedurale, Beschreibungsweise möglich ist, sollte das Vorgangsobjekt in der Lage sein, den Vorgang autonom abzuwickeln; anderenfalls wird eine Instanz des *Vorgangsmonitors* eingeschaltet, der mit den bereits eingeführten Problemlösungsstrategien eine finale Spezifikation in eine *afinale* überzuführen versucht; zu diesem Zweck stehen Kon-

sultationsregeln und ein Schnittstellenmodul zur Kommunikation mit dem Benutzer zur Verfügung. Die Kommunikation und Synchronisation zwischen Vorgangsobjekten findet über Nachrichtenaustauschmechanismen statt.

- Der Vorgangsmonitor entspricht mit seinen Instanzen den logischen Arbeitsplätzen im Büro. Ihm stehen neben den Konsultationsregeln sowie Vorgangs- und Datenobjekten noch Agenten zur Verfügung, die, ganz im Sinne der intelligenten Kommunikationswerkzeuge des Imail-Konzeptes (vgl. Abschnitt 2.2.4), die Kommunikation mit benachbarten Vorgangsmonitoren oder Vorgangsobjekten übernehmen. Als temporäres Kommunikationsmedium steht hierfür ein Arbeitsspeicher zur Verfügung.

ZDONIK84 beschreibt ein System, das ähnlichen Prinzipien folgt. Bemerkenswert an seinem Ansatz ist der Einbezug von Versionenkontrollmechanismen, die, analog zum bereits beschriebenen Standpunkt-Konzept, den Zustandsverlauf von Objekten abzubilden vermögen. Derartige Konzepte in der Datenbanktechnik führten bereits zu einer Vielzahl von Erweiterungen der Datenmodelle und Zugriffssprachen (vgl. dazu u.a. MÜLLER83).

Das Oz-Konzept (NIERSTRASZ85) bildet ein Objektmodell aus Inhaltsstücken und einer aus Prozedurfragmenten bestehenden Verhaltensbeschreibung. Der Ansatz wird im dynamischen Aspekt von Triggern dominiert, die auf bestimmte Ereignisse hin Handlungsschemata aktivieren, die durch Prozedurfragmente spezifiziert werden. Das entsprechende Vorlaufersystem ist in GIBBS83 skizziert.

Es ist bemerkenswert, daß in letzter Zeit in steigendem Maße versucht wird, auch herkömmliche Programmiersprachen mit Elementen der Objektorientierten Programmier Technik auszustatten.

Namentlich durch den Einzug von Methoden der Künstlichen Intelligenz und aufgrund der Insuffizienz der dabei propagierten Werkzeuge (LISP- und Prolog-Umgebungen) bei anwendungsnaher Programmierung wurde die Notwendigkeit weitergehender Reglementierung durch Aufbau übergeordneter Strukturen evident. Dabei ist der Objektorientierte Programmierstil, sofern er systemseitig unterstützt wird, ein probates Mittel zu Modellierung komplexer Sachverhalte und wird auch weiteren Einfluß auf zukünftige Entwicklungen haben.

4.3 TRANSAKTIONEN IM BÜRO

Dieser Abschnitt versucht eine Einordnung des Transaktionsbegriffs in die Büroumgebung, nachdem Transaktionen als atomare Ausführungseinheiten in Datenbank-Anwendungsprogrammen bereits anerkannt sind.

4.3.1 DER TRANSAKTIONSBEGRIFF

"A transaction is a sequence of DML-statements transforming the database from one logically consistent state to another logically consistent state." (HARDER83, S.42)

Diese Definition umreißt den Kern des sog. Transaktionsparadigmas, das im Zuge der Entwicklung des Experimental-systems R am IBM Research Laboratory in San Jose offengelegt wurde (vgl. hierzu ESWARAN76b und Abb. 4-12). Der Grundgedanke des Transaktionskonzeptes ist in GRAY81 (S. 144) durch die Ei-

enschaften *Consistency*, *Atomicity* und *Durability* fixiert worden, die dann in HARDER83 durch Aufspalten der Konsistenzaspekte zu dem Akronym ACID (*Atomicity*, *Consistency*, *Isolation*, *Durability*) umgeprägt wurde. Die Anforderungen bezeichnen im einzelnen (vgl. auch STEINBAUER83, S. 18ff):

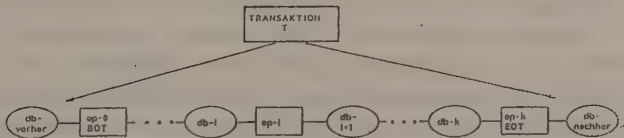


Abb. 4-12: Konstitution einer Transaktion aus elementaren DB-Operationen (aus STEINBAUER83, S. 22)

- **Atomizität (Atomicity):** Sie erzwingt alternativ entweder die vollständige Ausführung einer Transaktion oder das Rücksetzen aller ihrer Effekte. Der in STEINBAUER83 (S. 20) dabei genannte Zusatz "... in der Datenbank" legt die Gray'sche Definition (S. 144) zu eng aus und soll im folgenden keine Beachtung finden. Damit wird die Transaktion unteilbar; der Programmierer oder Auslöser einer Transaktion hat die Gewähr, daß keine intermediären Zustände des Systems nach Abschluß der Transaktion vorliegen. Jedweder Abbruch einer Transaktion muß entweder zu einem Rücksetzen auf den Anfangszustand vor Beginn der Transaktion (*Backward Recovery*) oder auf einen gültigen, und vorzugsweise den intendierten, Folgezustand (*Forward Recovery*) übergeführt werden. Durch die Forderung nach Atomizität

ist, nach der Terminologie der Handlungstheorie, die Komplexbildung aus Elementarhandlungen und anschließende Aktivierung des entstandenen Handlungsschemas möglich.

- **Konsistenz (Consistency):** Eine Transaktion überführt eine Datenbank aus einem konsistenten Vorzustand in einen konsistenten Nachzustand. Die gültigen Zustände werden dabei durch die deklarierten Integritätsbedingungen definiert, die vor und nach Ausführung einer Transaktion Gültigkeit haben müssen. Ähnliches gilt für die sog. *Übergangsbedingungen* (vgl. RÖHRLER, S. 15), die die Anwendung einer Transaktion durch bestimmte Vorbedingungen einschränken, aber im übrigen die gleichen Dienste leisten.

- **Isolation:** Diese oft auch als *Ablaufkonsistenz* bezeichnete Eigenschaft soll verhindern, daß durch die Einflüsse parallel ablaufender Transaktionen inkonsistente Zustände der Datenbank erzeugt und sichtbar werden. Der Ablauf einer Transaktion muß deshalb logisch völlig isoliert werden und keine Interferenzen mit anderen Transaktionen erzeugen; im Klartext heißt dies, daß eine Transaktion, sieht man von Verklemmungen und Zugriffskonflikten ab, im Rahmen der Modellvorstellung einer Einbenutzermaschine und -datenbank entworfen werden kann. Die Ablaufkonsistenz, d.h. die Zulässigkeit der durch die Transaktionen verursachten Wirkungen nach außen, wird durch das sog. *Serialisierbarkeitsprinzip* gewährleistet, das die Abbildbarkeit eines Transaktionsablaufs auf einen beliebigen seriellen Ablauf sicherstellt (*).

- *Dauerhaftigkeit*: Diese Anforderung stellt sicher, daß die Auswirkungen einer Transaktion nach deren Abschluß dahingehend abgesichert sind, daß durch Fehler (in Zentraleinheit und Peripherie) diese Auswirkungen erhalten bleiben und keine, ggf. ungewollte, Revision dieser Auswirkungen erfolgen kann. Eine Anullierung kann nur durch eine kompensierende Transaktion erfolgen. Mit dem ordnungsgemäßen Abschluß der Transaktion werden deren Auswirkungen für andere Transaktionen sichtbar und sind zur Weiterarbeit freigegeben.

Aufbauend auf diesen Anforderungen stellt sich der Ablauf der Modifikationen auf einem Datenbanksystem als der Ablauf einer Menge von Transaktionen dar, die nach STEINBAUER83 (S. 22) auch als *Transaktionssystem* bezeichnet werden kann.

4.3.2 NEUBEGRÜNDUNG DES ACID-KONZEPTE

Ziel dieses Abschnitts ist nun, das durch das ACID-Konzept begründete Anforderungssystem an den Erfordernissen der Bürowelt zu messen und ggf. zu ergänzen bzw. zu verwerfen.

Implizite Prämissen des ACID-Konzeptes. Im Zuge des Einsatzes

(*) Dieses Serialisierbarkeitsprinzip gewährleistet natürlich ausschließlich die Ablaufkonsistenz hinsichtlich bestimmter Anomalien (*Lost Update*, *Inconsistent View*, *Inconsistent Database*, *Phantom-Problem*). Alle Abhängigkeiten oberhalb des Transaktionsniveaus (Folgebedingungen) müssen organisatorisch oder über die üblichen Vorbedingungsmechanismen abgefangen werden.

von Transaktionssystemen, denen das ACID-Konzept vollständig oder in wesentlichen Teilen zugrundeliegt, haben sich folgende Prämissen herauskristallisiert, die größtenteils wohl von dem technologischen Stand beeinflusst wurden, der zur Zeit der Veröffentlichung des Konzeptes herrschte:

- (i) *Kurze und unterbrechungsfreie Transaktionen:* Das Umfeld der frühen Datenbanksysteme war durch kurze und unterbrechungsfrei ablaufende Transaktionen geprägt. Derartige Abläufe sind vor allem im betriebswirtschaftlichen Bereich (z.B. Buchungstransaktionen im Stapelbetrieb) typisch. Durch das Eindringen des Datenbankgedankens in die sog. *Nichtstandard-Bereiche* kann diese Annahme nicht mehr gehalten werden.
- (ii) *Frei formulierbare Transaktionen:* Der durch ACID konstituierte Transaktionsbegriff geht von der Annahme *Freier Transaktionen* aus. Dieser Begriff basiert auf der Klassifikation nach STEINBAUER83 (S. 121ff), die Transaktionen in folgende Kategorien einteilt:
- *Schematische Transaktionen*, deren Ablauf vollständig und deterministisch bestimmt ist und die derart einen funktionalen Übergang von einem Datenbankzustand in einen Folgezustand darstellen,
 - *Reglementierte Transaktionen*, deren Ablauf von Parametern, die erst zur Laufzeit bekannt werden, bestimmt ist, und
 - *Freie Transaktionen*, die alle Freiheiten der beliebigen Manipulation des Datenbestandes lassen und in keiner Weise, also auch nicht partiell, prädeterniniert sind.

Das Dilemma an dieser Stelle ist durch die implizite Annahme *Freier Transaktionen* im Hinblick auf Zugriffs-, Konsistenzüberwachungs- und Synchronisationsmechanis-

men, andererseits durch die gleichzeitige Implikation Schematischer Transaktionen hinsichtlich der Transaktionslaufzeit charakterisiert. Abgesehen von reinen Dialogumgebungen zur Eingabe von Ad-hoc-Anfragen, führt die erste Annahme zu hohem Aufwand bei Transaktionsabschluß zur Sicherstellung der Konsistenz und sehr restriktivem Verhalten hinsichtlich der Synchronisation; der zweite Aspekt impliziert eine Stapelumgebung mit feststehender Ablauffolge und sollte so gravierende Einschränkungen der zu Transaktionsende erforderlichen Konsistenzüberprüfungen ermöglichen.

(iii) *Hohes Isolationsbedürfnis*: ACID geht von der unabhängigen Operation diverser Transaktionen auf einem Datenbestand aus. Gerade im Umfeld des Büros sind aber neben der impliziten Synchronisation zwischen Abläufen bzw. Prozessen auch noch viele explizite Abhängigkeiten im Ablauf gegeben; gerade die Büroprozeduren, die den Gegenstand dieses Abschnitts bilden, erlauben eine Einschränkung der Isolationsanforderungen, da die Reihenfolge der Zugriffe sowie die Zugriffsmodi bekannt sind.

(iv) *Abgeschlossenheit des Systems*: Die Definitionen zur Atomizität und Dauerhaftigkeit gehen von der Abgeschlossenheit des Systems hinsichtlich seiner Auswirkungen auf die Umwelt aus. Im Klartext heißt dies, daß allein die Auswirkungen innerhalb der Datenbasis Gegenstand von Restaurierungs- bzw. Wiederholungsmaßnahmen zur Erzielung der geforderten Transaktionseigenschaften sind. Jim Gray kategorisiert die Aktionen, die die erwähnten Auswirkungen herbeiführen, in folgende drei Kategorien (GRAY81, S. 145):

- *Ungeschützte Aktionen:* Diese können in den folgenden Betrachtungen außer Acht gelassen werden, da ihre Auswirkungen per Definition als irrelevant und wiederholbar eingestuft worden sind. Beispiele für derartige Aktionen sind das Absenden einer unverbindlichen, informativen Zwischenmeldung, die keine unmittelbaren Folgen beim Empfänger hat, oder das Anlegen von temporären Dateien (Schmierzettel). Der letztgenannte Fall kann jedoch im Zusammenhang mit dem Wiederaufsetzen von Bearbeitungswerkzeugen an dem gerade erreichten Arbeitspunkt zu Problemen führen.
- *Geschützte Aktionen:* Bei jedwedem Transaktionsabbruch sind für geschützte Aktionen Recoverymaßnahmen möglich, da ihre Auswirkungen im System lokalisiert bleiben und so jederzeit (innerhalb der Transaktion) rücknehmbar sind. Als Beispiel ist hier die Modifikation von Objekten auf magnetischen Datenträgern zu nennen.
- *Realaktionen:* Nach Anstoßen der Aktion ist die Auswirkung nicht mehr unmittelbar rücknehmbar. Der Bezug zu den in Abschnitt 4.1.1 eingeführten *Intra-system-Handlungen* ist offenkundig, da in diesem speziellen Fall die Schnittstelle zwischen einem System immaterieller Objekte zu einem System materieller Objekte überschritten ist; jedoch ist diese Unterscheidung unwesentlich; signifikant ist nur das Protokoll zwischen den *kontrollierenden Instanzen*, das auch im Fall eines immateriell/immateriell-Überganges zur Behandlung der Übergangsaktion als Realaktion zwingen kann. Beispiele für Realaktionen sind das Absenden eines Briefes oder das Ausdrucken einer Liste am Rechner. Besondere Bedeutung haben Realaktionen in der Büroarbeit, da sehr viele der Aktionen Auswirkungen über die Systemgrenzen hinaus haben (Abwickeln von Briefverkehr, Ausdrucken von Urkunden etc.).

Nach der reinen ACID-Lehre müssen Realaktionen bis zum Transaktionsabschluß zurückgestellt werden (*). Da dies die Manövrierbarkeit in einem derartigen System, namentlich bei gleichzeitiger Ungültigkeit der Annahme (i), beträchtlich einschränken würde, ist hier ein bedeutender Kritikpunkt aus der Perspektive der Anforderungen im Büro zu sehen.

Welche Erweiterungen sind erforderlich? Die folgenden Ausführungen beziehen sich auf die Einbettung eines erweiterten Transaktionsbegriffs in ein Büroinformationssystem. Eine entsprechende Erweiterung bzw. Ergänzung des ACID-Konzeptes berührt die im folgenden angesprochenen Punkte. Dabei soll jedoch der Gesamtkontext, d.h. die beliebige Stornierung von Handlungskomplexen bzw. die Zusicherung ihrer atomaren, konsistenten und dauerhaften Durchführung, erhalten bleiben.

Einschränkung der Isolationsforderung. Die Transaktion im Büro ist mit dem Begriff der Bürohandlung gleichzusetzen. Wie in Abschnitt 4.1 angeführt, können Handlungen zu Komplexen vereinigt werden, die selbst wiederum Komplexe enthalten, und so die Konstitution von Bürohandlungen aus komplexen Handlungen erlauben. Analog hierzu gibt es eine Vielzahl von Veröffentlichungen zu dem Thema *geschachtelter Transaktionen* (stellvertretend hierfür KIM83 und WALTER85), die dieselbe Möglichkeit für Transaktionen unter Beibehaltung der typischen ACID-Eigenschaften ermöglichen. Beachtenswert ist, daß derart angelegte Transaktionen über extrem lange Zeiträume aktiv sein können (Wochen, Monate, sogar

(*) Jim Gray nennt hier auch das interessante und einprägsame Beispiel aus dem Betrieb von Geldausgabeautomaten: Die Wiedereinführung von Bargeld im Zuge der Rücknahme eines Ausgabevorgangs kann vom Bediener wohl nur schwerlich erwartet werden.

Jahre) und daß spezielle Protokolle die Teilhabe mehrerer Prozesse an den Datenressourcen ermöglichen müssen (Gruppentransaktion). Auch die gezielte Freigabe (*check-out*) einzelner Datenobjekte und die vorzeitige Aktivierung von Realaktionen, mithin also die Abkehr vom streng gehandhabten *Zwei-Phasen-Sperrprotokoll*, gehören zu den zu fordernden Eigenschaften dieser Transaktionen, wobei die Gefahr eventuellen rekursiven Zurücksetzens billigend in Kauf genommen werden muß.

Inverse von Aktionen. Zur Erhaltung der Atomizitäts- und Dauerhaftigkeitseigenschaften ist die spezielle Behandlung vorzeitig ausgeführter Realaktionen erforderlich. Es ist zu fordern, daß im Rahmen der Objektbeschreibung neben den am Objekt auszuführenden Elementarhandlungen gleichermaßen die Inversen dieser Handlungen zu deklarieren sind. So sollte zum Beispiel dem Absenden einer vorbehaltlichen, behördlichen Genehmigung auch eine Anweisung zum Versand eines Widerrufs beigelegt werden, um derart eine automatische Rücknahme dieser Maßnahme bewirken zu können. Die erforderlichen technischen Anpassungen wären z.B. in einem objektorientierten System durch Erweiterung des dortigen Methodenbegriffs leicht einzuführen.

Finale Transaktionsbeschreibung. Die REDO- und UNDO-Behandlung in Bürosystemen ist bei herkömmlichen Logging-Methoden im Grunde unsinnig, da die geforderte Eigenschaft der Flexibilität von Handlungsvorschriften im Büro, gerade im Falle finaler Beschreibung, durch Kontextänderung an anderer Stelle verlorengehen kann. Zu fordern ist deshalb die Aufnahme der Beschreibung der gewünschten Zielsituation, da nur so zu einem späteren Zeitpunkt, ggf. mit einer völlig anderen Handlungsfolge, der entsprechende Zielzustand hergestellt werden kann.

Qualitätsanforderungen. Allgemein ist zu bemerken, daß die absolut formulierten ACID-Anforderungen im skizzierten Rahmen nicht realisierbar sind. Ein Bürodatenbanksystem muß in höherem Maße kommunikativ sein, da es gewisse Leistungen, gerade bei der

Behandlung von Realaktionen, oft nur unvollständig erbringen kann, da Einschränkungen der Anwendungswelt, z.B. juristischer Natur, Zurücknahme- oder Restaurierungsoperationen nur begrenzt zulassen. Die Absolutheit der ACID-Forderungen muß der weitestgehenden Verwirklichung weichen.

Abschließend sollte noch bemerkt werden, daß die Anforderungen an Transaktionsverwaltungssysteme für Bürohandlungen unter Einschluß von Realhandlungen in diesem Rahmen nur tangential behandelt werden konnten und durchaus ein anspruchsvolles Thema für weitere Abhandlungen sein würden.

5 Stand der Technik und zukünftige Entwicklungen

Dieser abschließende Abschnitt führt einige Randbemerkungen zum derzeitigen Stand der Technik auf dem Bürosektor aus. Dabei werden auch einige Produkte namentlich Erwähnung finden, wobei jedoch aufgrund der Vielzahl von Anbietern explizit *keine* Vollständigkeit attestiert werden kann.

5.1 ÜBERBLICK ÜBER DERZEIT VERFÜGBARE BÜROSYSTEME

Der Markt für Bürosysteme wird derzeit immer noch von der Diskussion um die *Integration von Insellösungen* beherrscht; Preisfrage ist weniger die funktionale Anpassung oder begriffliche Normierung denn die datentechnische Anpassung und ggf. Übertragungsfähigkeit von Daten. So wird auch meist weniger von *Büroautomation* als von *Bürokommunikation* gesprochen, da, im Gegensatz zu den Betrachtungen dieser Abhandlung, der Kommunikationsaspekt, insbesondere die formalen Anforderungen der Kommunikation sowie die Übergabe der grundlegenden Kontextinformationen zur korrekten Interpretation, die überragende Rolle spielen.

Nun zu Anbietern und Produkten:

IBM. Die traditionelle Basis der IBM ist der Großrechnermarkt; aus diesem Grund sind die Angebote dieses Hauses auf zentralisierte Großlösungen ausgerichtet. Die Hauptprodukte sind dabei (vgl. GEHNE85)

- *PROFS (Professional Support)* in der VM/CMS-Umgebung und
- *DISSOS (Distributed Office Support System)* in der MVS-Umgebung.

Diese Systeme bieten Erstellungs- und Ablagehilfen für Textdaten sowie die Möglichkeit von Recherchen und der Integration von Text- und Datenverarbeitung. Die Basis bildet die bereits erwähnte Alternative zu ODA, der DCA-Standard. Weitergehende Konzepte zur Vorgangsbearbeitung sind dabei nicht vorgesehen.

Ein weiterer Ansatzpunkt ist die Integration der zentralen mit der lokalen Verarbeitungskapazität, d.h. der PC/Mainframe- bzw. Terminal/Mainframe-Integration. In diesem Markt sind auch OEM-Anbieter aktiv; ein Beispiel dafür beschreibt BEHRINGER85. Der Markt der Abteilungsrechner im Minirechnerformat, den das System /36 abdecken sollte, kann aufgrund der Hard- und Softwarearchitektur dieses Systems von IBM nicht befriedigend bedient werden.

Xerox. Das bereits mehrfach erwähnte System 8010 ist der Kern einer Familie von Arbeitsplatzrechnern und Peripheriegeräten, die über ein lokales Netzwerk (Ethernet) integriert sind. Durch Gateways ist der Übergang in die SNA-Welt der IBM-Großrechner möglich. Signalwirkung für den Wettbewerb hatten die Gestaltungsgrundsätze der Benutzer- und Werkzeugschnittstelle, auf die bereits mehrmals hingewiesen wurde (u.a. durch SMITH82). Nachteilig für die Vermarktung des Systems wirkte sich die weitgehende Untauglichkeit der Soft- und Hardware im Hinblick auf die Integration von Drittprodukten aus.

Interleaf. Dieser Nachteil wurde von Interleaf, Inc. durch die Entwicklung der Produkte WPS (*Workstation Publishing System*) und TPS (*Technical Publishing System*), die auf All-Purpose-Arbeitsplatzrechnern der Firmen Apollo, Sun, DEC, PCS und IBM mit anderen Programmpaketen integriert werden können, behoben. Dennoch ist die von dieser Produktlinie verwendete Dokumentenarchitektur in hohem Maße an den Funktionsvorrat des jeweiligen Bearbeitungspaketes gekoppelt. Die Bearbeitungsmöglichkeiten für nichttextuelle Datentypen sind eingeschränkt; weder die Bearbeitung zwei- oder dreidimensionaler grafischer Objekte noch die rasterisierter Bilder entspricht dem auf diesen Gebieten zu fordernden Standard. Durch die Möglichkeit eines Übergangs von grafischen Standardformaten in die WPS-interne Geometriedarstellung kann die repräsentative ER-Kompatibilität (vgl. Abschnitt 2.3) zu anderen Bearbeitungswerkzeugen geschaffen werden.

Siemens. Das Angebot der Siemens AG erstreckt sich in weiten Zügen auf die Produkte anderer Hersteller, die entweder in Lizenz hergestellt werden oder vom Hause Siemens als OEM vertrieben werden. Erwähnenswert ist dabei das System EMS 5800, das baugleich dem Xerox System 8010 mit Adaptionen für den deutschen Markt offeriert wird. Ein neueres Produkt auf dem Textverarbeitungssektor bietet das Siemens mit dem System WS30 (baugleich der Apollo-Linie) und dem WPS-Paket von Interleaf an.

DEC. Die DEC-Systeme sind traditionell am technisch-wissenschaftlichen Bereich orientiert. In jüngerer Zeit hat sich diese Orientierung eher dem Bedarf des nicht-technischen Endbenutzers angepaßt. Das zentrale Programmpaket ist *Alles-in-1* (vgl. DIGITAL85), das für die Rechner der VAX-Klasse und die verschiedenen angebotenen Arbeitsplatzrechner (*MicroVAX*, *DECmate*, *Professional-300* und *Rainbow-100*) verfügbar ist. Integriert in dieses Paket ist ein Textverarbeitungssystem, das auf Rechnern aller Klassen funktionsgleich lauffähig ist.

Wang. Dieser Anbieter ist mit dem Produkt Wang OFFICE, das auf Abteilungsrechnern lauffähig ist, auf dem Markt vertreten. Wang bietet eine glückliche Kombination von lokalen und zentralen Verarbeitungsmöglichkeiten, wobei der Zugriff auf den Zentralknoten transparent vom Lokalrechner aus erfolgt. Arbeitsplatzrechner der PIC-Klasse bieten Rastengrafik mit für den Bürobereich weit entwickelten Grafikfähigkeiten an, wobei jedoch auch hier nur die repräsentative ER-Kompatibilität auf Rasterebene angeboten wird. Der Text- und Kommunikationsbereich ist durch ein fortgeschrittenes Textsystem (WP Plus) sowie diverse audio-visuelle Kommunikationsmöglichkeiten über ein Breitbandnetz sowie Low-Cost-Verbindungen und Gateways in die IBM-Kommunikationsarchitektur abgedeckt.

Data General. Data General betont die Integration von Rechen-systemen sowohl auf Hardware- als auch auf Softwareebene. Die Integration verschiedener Betriebssysteme erlaubt den Einsatz eines breiten Angebots an Drittsoftware. Der allgemeine Rahmen für Büroautomationsaktivitäten wird durch das System CEO (*Comprehensive Electronic Office*) (vgl. SCHOLL85) gesetzt, das Textverarbeitungs- und Kommunikationsaktivitäten (auch mit Stimmverarbeitung) unterstützt.

Apple. Diesem Anbieter gebührt das Verdienst, hochentwickelte Schnittstellenkonzepte auf extrem preisgünstigen Maschinen anbieten zu können (*Apple Lisa*, *Apple Macintosh*). Allerdings ist die Anschlußfähigkeit an andere Systeme stark beschränkt, gleichermaßen ist die Integration von Eigen- und Drittsoftware nur in eingeschränktem Maße möglich. Deshalb ist Apple weniger im investitionsintensiven professionellen Bürobereich erfolgreich als im Einzelgeschäft mit Freiberuflern und Klein-gewerbetreibenden.

Abschließend muß nochmals betont werden, daß der Büroautomationsmarkt von der funktionalen Integration verschiedener Werkzeuge und den Kommunikationsproblemen innerhalb des Hauses sowie über öffentliche Netze beherrscht wird. Keines der genannten Produkte bietet befriedigende Lösungen für die Vorgangssteuerung und -überwachung an; die Integration verschiedener Datentypen in einem übergeordneten Dokumentenmodell ist höchstens Fernziel. Alle Systeme sind eher bemüht, durch graduelles Abwerten von Bearbeitungsfunktionen und die Übertragung auf billige Datenendgeräte den Durchschnittspreis für einen Arbeitsplatz zu drücken und so die Investitionsbereitschaft in diesem Markt zu fördern. Diese Tendenz, die aufgrund des technischen Fortschritts nur zeitweiligen Bestand haben kann, ist infolge der dadurch angesammelten "Schnittstellenhypothesen" für die Zukunft eher hinderlich denn förderlich, zumal dadurch auch der Erwartungshorizont der Benutzer für zukünftig verfügbare Systeme gesenkt wird.

5.2 STAND DER DATENBANKTECHNIK

Der Bürobereich ist eines der zentralen Zielgebiete für den Einsatz der sog. *Nichtstandard-Datenbanksysteme*, die seit einigen Jahren die Forschungsaktivitäten dieses Bereichs beherrschen. Die Annahmen, die dem Einsatz herkömmlicher Datenbanksysteme zugrundeliegen, sind in Abschnitt 4.3.2 bezüglich des Transaktionsgedankens angeführt; weiterhin sind noch die in diesen Datenbanksystemen vorherrschende *einfache Struktur der Datenobjekte* sowie die *zentrale Datenhaltung* zu nennen (vgl. KÜSPERT86, S. 184).

Neue Anforderungen im Nichtstandard-Bereich sind im folgenden angeführt:

- *Verwaltung großer & komplexer Objekte:* Die abzulegenden Internen Repräsentationen setzen sich aus einer Vielzahl von Teilobjekten zusammen und sind gemeinhin von großem Umfang. Die dabei auftretenden Strukturen sind oft in mehrstufigen Hierarchien angeordnet oder sogar rekursiv; Beispiele hierfür nennt LAMERSDORF⁸⁴; die Unzulänglichkeit der herkömmlichen Datenbanksprachen (namentlich der relationalen) belegt AHO⁷⁹.
- *Komplexe Integritätsbedingungen:* Die Objektstruktur wird mit Hilfe von nicht-trivialen Integritätsbedingungen konstituiert, die sich nur schwer und unter Berücksichtigung der Speicherungsstruktur überwachen lassen.
- *Lange Transaktionen:* Dieser Punkt wurde bereits in Abschnitt 4.3 erörtert.
- *Versionenkontrolle:* Zur Unterstützung der Objektverwaltung muß der Faktor *Zeit* hinsichtlich der Veränderung der Objekte mitberücksichtigt werden.
- *Verteilung:* Die Architektur derartiger Systeme (Zentraler Dateiverwaltungsknoten mit lokaler Verarbeitungskapazität und Speicherfähigkeit) fordert Verteilte Datenbanken mit verschiedenen abgestuften Protokollen der Zugriffssynchronisation.

Die Forschungsaktivitäten konzentrieren sich dabei auf die folgenden Teilbereiche:

- *Datenmodell*: Die hier vorliegenden Vorschläge sind Legion; man kann nur konstatieren, daß die Illusion eines allgemein im Nichtstandardbereich einzuführenden Datenmodells sich als eine solche erwiesen hat (*). Durch die Erkenntnis, daß mit rein strukturellen Maßnahmen den Anforderungen des Nichtstandard-Bereichs nicht beizukommen ist, hat sich das große Feld der Forschungen auf dem Gebiet der (mehr oder weniger) objekt-orientierten Datenbanksysteme herauskristallisiert (vgl. Abschnitt 4.2.3).
- *Interne Repräsentation*: Die Interne Repräsentation wurde im Zuge der Entwicklung neuer Datenstrukturen und Indexierungsverfahren um diverse Neuerungen erweitert. Exemplarisch seien hier nur die *Gridfile-Speichertechnik* zur Darstellung und Wiedergewinnung räumlicher Daten (vgl. HINRICHS83) oder die *Signaturverfahren* zur Invertierung von Freitext (vgl. FAOUTSOS85) genannt.
- *Handhabung von Objekten*: Dabei sind insbesondere die Objektverwaltung durch neue Sprachschnittstellen, die Integritätserhaltung durch Triggermechanismen und die Verwaltung von Kontextwissen für die Abfragen angesprochen.
- *Synchronisation*: Durch die Erweiterung der strikten Synchronisationsmechanismen herkömmlicher Datenbank-

(*) Man denke nur an die Erweiterung des relationalen Datenmodells um nichtatomare Datenelemente im Zuge der Entwicklung des NF2-Datenmodells (vgl. u.a. DADAM86). Dieses Modell zeigt zwar durchaus angenehme Strukturierungseigenschaften für die abzubildenden Datenelemente; der dynamische Aspekt bleibt jedoch den Schwächen der herkömmlichen Datenbanksysteme verhaftet.

systeme um qualitativ-abgewertete Zugriffsmöglichkeiten (z.B. *Snapshot*) ist neben der Verbesserung des Zugriffsverhaltens auch die Möglichkeit der Definition langer Transaktionen gegeben. Weitere Aktivitäten richten sich auf die Einführung von *Gruppentransaktionen*, bei denen die Synchronisation und deshalb auch die Wiederherstellung im Fehlerfall teilweise den Anwendern der Datenbank überlassen wird.

Zur Architektur von Nichtstandard-Datenbanksystemen sind mehrere Ansätze denkbar, die in HÄRDER85 auf folgende vier Kategorien zurückgeführt wurden (vgl. auch Abb. 5-1):

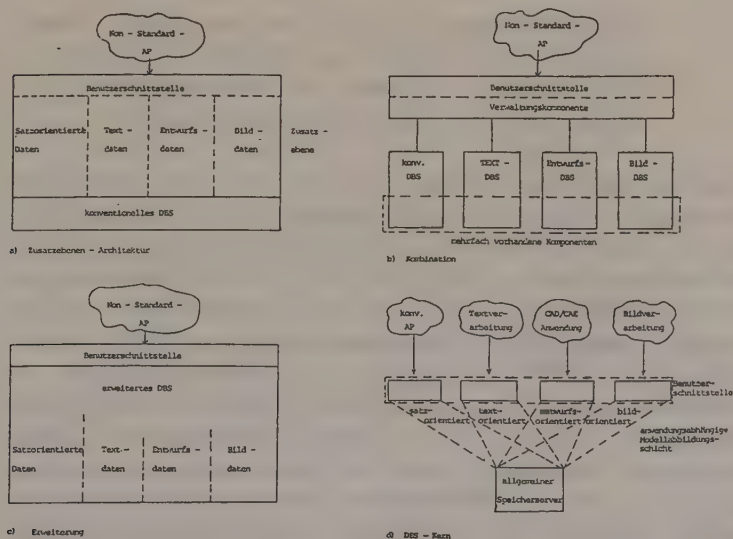


Abb. 5-1: Architekturvorschläge für Nichtstandard-Datenbanksysteme (aus HÄRDER85, S. 260)

- *Zusatzebenen-Architektur*: Ein konventionelles Datenbanksystem wird um eine Nichtstandard-Schicht erweitert.
- *Hybrid-Architektur*: Eine allgemeine Datenbank wird mit dedizierten Teil-Datenbanken bei Benutzung gemeinsamer Komponenten verbunden.
- *DBS-Erweiterung*: Hierbei findet eine Erweiterung des Datenbanksystems sowohl horizontal um neue Datentypen wie auch vertikal um weitere Funktionen für jeden Datentyp statt.
- *Kern-Datenbanksystem*: Ein Basismechanismus bietet grundlegende Datenverwaltungsfunktionen an; darauf setzen dedizierte Schichten zur Implementierung komplexerer Operatoren für jeden Datentyp ein.

Obwohl HARDER85 das Kern-Datenbanksystem als unter den gegebenen Verhältnissen optimal propagiert, erscheinen die DBS-Erweiterungen im Sinne der objektorientierten Datenmodelle doch am erfolgversprechendsten zu sein. So verführerisch der Gedanke des Kern-Datenbanksystems auch sein mag, bedeutet er doch die partielle Aufgabe der Datenbankidee an sich. Man kann an dieser Stelle nur feststellen, daß sich die Datenbanktechnik am Scheideweg befindet: Entweder werden sich die weiteren Aktivitäten auf die Einführung weiterer Zugriffstechniken bzw. deren Raffinement beschränken, oder man kommt durch eine Übernahme großer Teile der Anwendungsprogrammierung zu einer Vereinnahmung anderer Bereiche der Informatik. Die Erweiterungsarchitektur (etwa im Sinne der objektorientierten Erweiterung) erlaubt die Beschränkung auf Basistechniken (z.B. Nachrichtenübermittlung als Basistechnik für die Kommunikation zwischen Objekten), ohne die alle weiteren Arbeiten auf diesem Gebiet in der Vielzahl der zu behandelnden Detailprobleme versinken würden.

5.3 STAND DER KOMMUNIKATIONSTECHNIK

Die Situation in der Kommunikationstechnik ist seit der Einführung des ISO/OSI-Standards für offene Systeme (vgl. TANENBAUM⁸¹, S. 15ff) sowie der angeschlossenen Substandards in hohem Maße überschaubar geworden. Abb. 5-2 zeigt die Grundzüge dieser Schichtenarchitektur im Falle der Vermittlung von Endknoten zu Endknoten über Zwischenknoten.

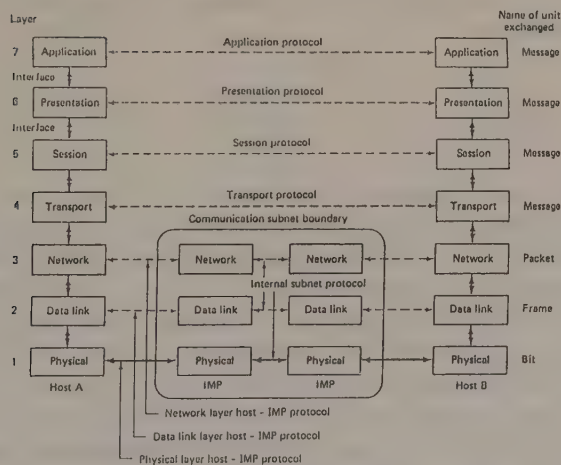


Abb. 5-2: Schichtenarchitektur eines Kommunikationssystems
(aus TANENBAUM⁸¹, S. 16)

In der Burokommunikation sind folgende Forschungsgebiete auf dem Kommunikationssektor aktuell:

- *Standardisierung höherer Dienste:* Hier hat namentlich der bereits erwähnte X.400-Standard mit den entsprechenden Substandards Einfluß gewonnen. Die Normen zum Dokumentenaufbau liefern hier Zubringerdienste, haben jedoch nicht ausschließlich Einfluß auf den Kommunikationssektor, sondern auf die rechnerinterne Datenorganisation an sich (vgl. u.a. TILLINGER86).

- *Parallele Erbringung heterogener Dienste:* Hier hat in der Bundesrepublik insbesondere der ISDN-Standard Einfluß genommen, der herkömmlichen Fernsprechsyste men den parallelen Betrieb unterschiedlicher Systeme und die Erbringung entsprechender Dienste erlaubt (vgl. KÜHN86).

5.4 ZUR ROLLE DER KÜNSTLICHEN INTELLIGENZ IN DER BÜROAUTOMATION

"AI is whatever hasn't been done yet."

(Larry Tesler in HOFSTADTER80)

"Artificial Intelligence has the same relation to intelligence as artificial flowers have to flowers. From the distance they may appear much alike but when closely examined they are quite different."

(D.L. Parnas in PARNAS85)

Über den positiven Einfluß der Techniken der Künstlichen Intelligenz auf fortgeschrittene Bürosysteme ist in Abschnitt 4 bereits ausführlich bezuggenommen worden. Außer in der Ablaufsteuerung sind auch in dem nur kurz gestreiften Bereich der Dokumentenanalyse (nähere Ausführungen hierzu in MERTENS86, S. 254, ausführlich in SCHUMANN86, S. 66ff) sowie in der Gestaltung von Benutzerschnittstellen diese Wirkungen spürbar. Man muß sich jedoch immer vergegenwärtigen, daß keiner dieser Einflüsse an bestimmte Systeme oder Softwareumgebungen gebunden ist, daß mithin der Mythos um bestimmte Programmiersprachen in diesem Bereich nicht die *Machbarkeit* eines Büroinformationssystems, sondern den *Komfort* im Zuge seiner Erstellung betrifft, wobei aufgrund der derzeit starken Niveauunterschiede zwischen forschungsorientierten Prototypsystemen und den im Feldeinsatz befindlichen Paketen dem Laufzeitaspekt nur untergeordnete Bedeutung beigemessen wird. Entscheidend für den Einsatz ist die Beschränkung der Daten- oder Wissensorganisation sowie -interpretation auf einen bestimmten problemadäquaten Stil, der

durch den Aufbau auf einer problembezogenen Programmierungsumgebung unterstützt werden kann. Derartige *Expertensystem-Shells* (vgl. u.a. BOBROW86) bieten eine vorgeformte Umgebung für das Wissen des Anwendungsbereichs, können also auch für den Bürobereich speziell zugeschnitten werden. Über derartige Versuche berichtet u.a. EDER86.

5.5 DIE ZUKUNFT

*"Work expands so as to fill the time
available for its completion."*

(C.N. Parkinson in PARKINSON65, S. 11)

Es ist schwierig, einen langfristig vorgezeichneten Weg weiter extrapolieren zu wollen. Die derzeitige Situation im Feld der Büroautomation zwingt wohl in den nächsten Jahren eher zu Anpassungsarbeiten und Feldversuchen denn zur Erarbeitung wiederum neuer Konzepte, die, wie die bereits vorhandenen, nur an einigen wenigen, oft unrealistischen Beispielen validiert werden. Die grundsätzliche Veränderung in der Büroautomation betrifft in weiten Zügen die Loslösung vom Primat des Applikationswerkzeugs, wie dies in Abschnitt 2 bereits angedeutet wurde, und die Hinwendung zu autonomen Datenstrukturen und deren Verselbständigung in Zeit und Raum. Diese Vision ist u.a. von Tsiichritzis und Nierstrasz (TSICHRITZIS85b, TSICHRITZIS85c) durch den skizzenhaften Aufbau von Welten initiativer Wissensträger (knos) gestützt worden. Diese Seite der Fortentwicklung soll jedoch in keiner Weise die zukünftigen Entwicklungen in der Hardwareentwicklung (höher auflösende Bildschirme, schnellere Prozessoren, sicherere Übertragungswege) und in der Entwicklung

für Basiswerkzeuge schmälern. Entscheidend wird jedoch sein, mit welchem Nachdruck sich die ersten Lösungen, die diesen Namen verdienen, bei Großanwendern für den weiteren Einsatz und die Fortentwicklung empfehlen.

Der entscheidende Faktor, der implizit bereits in die Betrachtungen des ersten und vierten Abschnitts Eingang gefunden hat, ist die hinreichende Adaptivität des Systems an organisatorische Veränderungen der betrachteten Unternehmenseinheit. Erfahrungsgemäß entsteht der überwiegende Kostenanteil kaum im Zuge des Entwurfs und der Erstellung eines Büroinformationssystems, eher denn während der Veränderung bzw. Anpassung an geänderte und neu eingeführte Arbeitsabläufe. Abb. 5-3 zeigt schematisch den schrittweisen Übergang eines Organisationszustandes O auf O' , wobei der eigentlich parallel mitzuziehende Systemzustand S meist im Ursprungszustand verharret, bis die auftretenden Diskrepanzen eine sprunghafte Anpassung, d.h. eine Globalrevision, erforderlich machen.

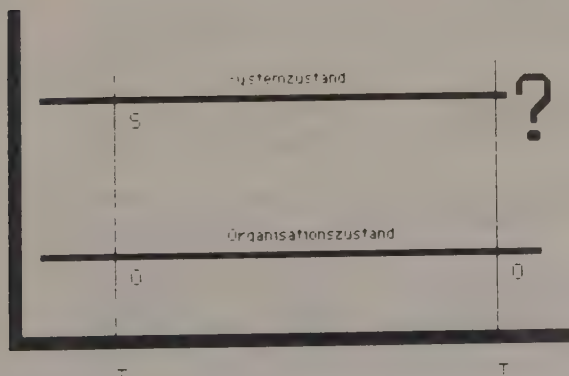


Abb. 5-3: Veränderung der Organisationsstruktur

Die Anstrengungen beim Entwurf von Basismechanismen zur Ablaufbeschreibung müssen darauf gerichtet sein, den graduellen Übergang beherrschbar zu machen, wobei namentlich Versionenkonzepte für Ablaufbeschreibungen eine entscheidende Rolle spielen dürften. C.N. Parkinson stellt in seinen Untersuchungen des dynamischen Verhaltens von Organisationen fest, daß das starre Festhalten an fixierten Organisationsstrukturen (in seinem Falle durch das Gebäude manifestiert, das die Organisation beherbergt) den eventuell langsamen, aber dennoch sicheren Tod der Organisation bewirkt. So soll denn seine Schlußfolgerung auch den Abschluß dieser Abhandlung bilden:

"... experience proves that such an institution will die. It is choked by its own perfection. It cannot take root for lack of soil. It cannot grow naturally for it is already grown. Fruitless by its very nature, it cannot even flower. When we see an example of such planning ... the experts among us shake their heads sadly, draw a sheet over the corpse, and tiptoe quietly into the open air."

(PARKINSON65, S. 84f)

Anhang A

Literaturverzeichnis

- ABRAHAM81 Abraham, S.: The Impact of Automated Systems on the Productivity of Managers and Professionals, Proc. of the 1981 Office Automation Conference, Houston 1981
- AHLSSEN84 Ahlsen, M. et al.: An Architecture for Object Management in OIS, ACM Transactions on Office Information Systems 2 (1984) 3, S. 173ff
- AHO79 Aho, A.V.; Ullman, J.D.: Universality of Data Retrieval Languages, Proc. of the 6th Annual ACM Symposium on Principles of Programming Languages, San Antonio, Texas, Jan. 1979
- ARIAV85 Ariav, G.; Ginzberg, M.J.: DSS Design: A Systematic View of Decision Support, Communications of the ACM 28 (1985) 10, S. 1045ff
- ATTEWELL84 Attewell, P.; Rule, J.: Computing and Organizations: What We Know And What We Don't Know, Communications of the ACM 27 (1984) 12, S. 1184ff
- BALZERT85 Balzert, H.; Hoppe, H.U.; Ziegler, J.: Fenster-systeme im Vergleich, Forschungsberichte WISDOM-Verbundprojekt TRIUMPH-ADLER AG, Nürnberg 1985
- BARBER82a Barber, G.: Embedding Knowledge In A Workstation, in: Naffah, N.: Office Information Systems, North-Holland, Amsterdam 1982
- BARBER82b Barber, G.; Hewitt, C.: Foundations for Office Semantics, in: Naffah, N.: Office Information Systems, North-Holland, Amsterdam 1982

- BARBER83 Barber, G.: Supporting Organizational Problem Solving with a Work Station, ACM Transactions on Office Information Systems 1 (1983) 1, S. 45ff
- BARCOMB81 Barcomb, D.: Office Automation - A Survey of Tools and Technology, Digital Press 1981
- BEHRINGER85 Behringer, E.J.: Ein Nachrichtensystem für die Büroorganisation eines Großbetriebes, Diplomarbeit Universität Erlangen-Nürnberg IMMD VI, 1985
- BERNAL82 Bernal, M.: Interface Concepts For Electronic Forms Design And Manipulation, in: Naffah, N.: Office Information Systems, North-Holland, Amsterdam 1982
- BLUMENFELD85 Blumenfeld, M.: Dokumentarchitektur und Austauschformate, in: Wedekind, H.; Kratzer, K. (Hrsg.): Büroautomation '85, Teubner, Stuttgart 1985
- BOBROW86 Bobrow, D.G.; Mittal, S.; Stefik, M.J.: Expert Systems: Perils and Promise, Communications of the ACM 29 (1986) 9, S. 880ff
- BONO85 Bono, P.R.: A Survey of Graphics Standards and Their Role in Information Interchange, IEEE Computer 18 (1985) 10, S. 63ff
- BORGIDA85 Borgida, A.: Language Features for Flexible Handling of Exceptions, ACM Transactions on Database Systems 10 (1985) 4, S. 565ff
- BORNING82 Borning, A.H.; Ingalls, D.H.H.: Multiple Inheritance in Smalltalk-80, in: Waltz, D. (Hrsg.): Proceedings of the AAAI National Conference on Artificial Intelligence, Pittsburgh 1982
- BRÜER84 Brüer, J.-O. et al.: Office Models, ASLAB MEMO 84-1, Universitetet och Tekniska Högskolan Linköping, Linköping 1984
- BULL86 Bull, H.P.: Thesen zu den sozialen und rechtlichen Risiken der Informationstechnik, Informatik und Recht 1 (1986) 1, S. 3ff
- CARBONELL83 Carbonell, J.G.: User Modelling and Natural Language Interface Design, in: Balzert, H. (Hrsg.): Software Ergonomie, Tagungsband, Teubner, Stuttgart 1983
- CARDELLI85 Cardelli, L.; Wegner, P.: On Understanding Types, Data Abstraction, and Polymorphism, ACM Computing Surveys 17 (1985) 4, S. 471ff

- CHAMBERLIN82 Chamberlin, D.D. et al.: JANUS: An Interactive Document Formatter Based On Declarative Tags, IBM Systems Journal 21 (1982) 3, S. 250ff
- CHAMBERLIN85 Chamberlin, D.D.: Document Convergence in an Interactive Formatting System, IBM Research Report RJ 4916, Dezember 1985
- CLOCK SIN81 Clocksin, W.F.; Mellish, C.S.: Programming in Prolog, Springer, New York 1981
- COHEN86 Cohen, E.S.; Smith, E.T.; Iverson, L.A.: Constraint-Based Tiled Windows, IEEE Computer Graphics and Applications 6 (1986) 5, S. 35ff
- COLEMAN83 Coleman, W.T.; Warren, S.: The VisiOn Operating Environment, Database Engineering 6 (1983) 3, S. 21ff
- COPELAND84 Copeland, G.; Maier, D.: Making Smalltalk a Database System, SIGMOD Record 14 (1984) 2, S. 316ff
- COUTAZ84 Coutaz, J.: A Paradigm For User Interface Architecture, Carnegie-Mellon University, Pittsburgh PA, Research Report CMU-CS-84-124
- COUTAZ85 Coutaz, J.: Abstractions for User Interface Design, IEEE Computer 18 (1985) 9, S. 21ff
- CURRY83 Curry, G.A.: An Overview Of The Architecture(s) Of The Xerox Star Office System, Database Engineering 6 (1983) 3, S. 12ff
- DADAM86 Dadam, P. et al.: A DBMS Prototype to Support Extended NF2 Relations: An Integrated View on Flat Tables and Hierarchies, in: Zaniolo, C.: Proceedings of ACM SIGMOD '86, Washington, D.C. 1986
- DELISLE86 DeLisle, N.; Schwartz, M.: Neptune: A Hypertext System for CAD Applications, in: Zaniolo, C.: Proceedings of ACM SIGMOD '86, Washington, D.C. 1986
- DEMILLO79 DeMillo, R.A.; Lipton, R.J.; Perlis, A.J.: Social Processes and Proof of Theorems, Communications of the ACM 22 (1979) 5, S. 271ff
- DESOUZA81 DeSousa, M.R.: Electronic Information Interchange in an Office Environment, IBM System Journal 22 (1981) 1, S. 4ff
- DICKSON86 Dickson, G.W.; DeSanotis, G.; McBride, D.J.: Understanding the Effectiveness of Computer Graphics for Decision Support - A Cumulative Experimental Approach, Communications of the ACM

29 (1986) 1, S. 40ff

- DIGITAL85 Digital Equipment GmbH: Alles-in-1 Büro-Informationssystem - Handbuch (Firmenveröffentlichung), München 1985
- DIN84 Deutsches Institut für Normung: DIN 66234 / Teil 8: Bildschirmarbeitsplätze - Grundsätze der Dialoggestaltung, Beuth, Berlin 1984
- DISSA86 Di Sessa, A.A.: Notes on the Future of Programming: Breaking the Utility Barrier, Notizen zu interaktiven Systemen o. Jgg. (1986) 15, S. 51ff
- DITTRICH85 Dittrich, K.R.; Lorie, R.A.: Object-Oriented Database Concepts for Engineering Applications, IBM Research Report RJ 4691, August 1985
- DODSWELL83 Dodswell, A.: Office Automation, John Wiley, 1983
- DÖRNER79 Dörner, D.: Problemlösen als Informationsverarbeitung, 2. Auflage, W. Kohlhammer, Stuttgart 1979
- DRISCOLL82 Driscoll, J.: Office Automation: The Dynamics of a Technical Boondoggle, in: Landau, R.M.; Blair, J.H. (Hrsg.): Emerging Office Systems, Norwood, N.J. 1982
- DÜRRE83 Dürre, B.; Geldmacher, W.; Streppel, H.; Schnupp, P.: Objektorientierte Konstruktion interaktiver Werkzeuge, in: Stoyan, H.; Wedekind, H. (Hrsg.): Objektorientierte Soft- und Hardwarearchitekturen, Teubner, Stuttgart 1983
- EDER86 Eder, J. et al.: Informationssysteme für Klein- und Mittelbetriebe, in: Schulz, A.: Die Zukunft der Informationssysteme - Lehren der 80er Jahre, Springer, Berlin 1986
- EFFELSBURG85 Effelsberg, W.; Lamersdorf, W.: Strukturierung von Büroaktivitäten, in: Wedekind, H.; Kratzer, K. (Hrsg.): Büroautomation '85, Teubner, Stuttgart 1985
- ELLIS80a Ellis, C.A.; Nutt, G.J.: Office Information Systems and Computer Science, ACM Computing Surveys 12 (1980) 1, S. 27ff
- ELLIS80b Ellis, C.A.; Gibbons, R.; Morris, P.: Office Streamlining, in: Naffah, N.: Integrated Office Systems - Burotics, Proceedings of the INRIA/IFIP TC-6 International Workshop on Integrated Office Systems - Burotics, North-Holland, Amsterdam 1980

- ELLIS82a Ellis, W.: From the Editors, Office: Technology and People 1 (1982) 1
- ELLIS82b Ellis, C.A.; Bernal, M.: OFFICETALK-D: An Experimental Office Information System, Proceedings of the ACM SIGOA Conference on Office Information Systems, ACM, New York 1982
- ELLIS83 Ellis, C.A.: Formal and Informal Models of Office Activity, in: Mason, R.E.A. (Hrsg.): Proceedings of the IFIP 9th World Computer Congress, North-Holland, Amsterdam 1983
- ENDERLE83 Enderle, G.; Kansky, K.; Pfaff, G.; Prester, F.-J.: Die Funktionen des Graphischen Kernsystems, Informatik-Spektrum 6 (1983) 2, S. 55ff
- ESWARAN76a Eswaran, K.P.: Specifications, Implementations and Interactions of a Trigger Subsystem in an Integrated Database System, IBM Research Report RC 1820, November 1976
- ESWARAN76b Eswaran, K.P.; Gray, J.N.; Lorie, R.A.; Traiger, I.L.: On the Notions of Consistency and Predicate Locks in a Data Base System, Communications of the ACM 19 (1976) 11, S. 624ff
- FALOUTSOS85 Faloutsos, C.: Access Methods for Text, ACM Computing Surveys 17 (1985) 1, S. 49ff
- FIKES80 Fikes, R.E.; Henderson, D.A.: On Supporting the Use of Procedures in Office Work, in: Balzer, R.M. (Hrsg.): Proc. of the 1st Annual National Conference on Artificial Intelligence, Stanford 1980
- FIKES81 Fikes, R.E.: Odyssey: A Knowledge-Based Assistant, Artificial Intelligence 16 (1981) 3, S. 331ff
- FISCHER83 Fischer, G.: Entwurfsrichtlinien für die Software-Ergonomie aus der Sicht der Mensch-Maschine-Kommunikation, in: Balzer, H. (Hrsg.): Software Ergonomie, Tagungsband, Teubner, Stuttgart 1983
- FOLEY82 Foley, J.D.; Van Dam, A.: Fundamentals of Interactive Computer Graphics, Addison-Wesley, Reading MA 1982
- FOLEY84 Foley, J.D.; Wallace, V.L.; Chan, P.: The Human Factors of Computer Graphics Interaction Techniques, IEEE Computer Graphics and Applications 4 (1984) 11, S. 13ff

- FRIEDRICH86 Friedrich, V.: Einbettung von Legenden in eine INGRES-Datenbankumgebung, Studienarbeit Universität Erlangen-Nürnberg IMMD VI, 1986
- FURUTA82 Furuta, R.; Scofield, J.; Shaw, A.: Document Formatting Systems: Survey, Concepts and Issues, ACM Computing Surveys 14 (1982) 3, S. 417ff
- GEHNE85 Gehne, W.: IBM Bürokommunikation, in: Wedekind, H.; Kratzer, K. (Hrsg.): Büroautomation '85, Teubner, Stuttgart 1985
- GETHMANN79a Gethmann, C.F.: Protologik - Untersuchungen zur formalen Pragmatik von Begründungsdiskursen, Suhrkamp, Frankfurt (Main) 1979
- GETHMANN79b Gethmann, C.F.: Zur formalen Pragmatik der Normenbegründung, in: Mittelstraß, J.: Methodenprobleme der Wissenschaften vom gesellschaftlichen Handeln, Suhrkamp, Frankfurt (Main) 1979
- GIBBS83 Gibbs, S.; Tsichritzis, D.: A Data Modeling Approach for Office Information Systems, ACM Transactions on Office Information Systems 1 (1983) 4, S. 299ff
- GLADNEY85 Gladney, H.M.; Bennett, J.L.: Object Models for Office Systems -- An Exposition of Architectural Concepts, IBM Research Report RJ 4584, San Jose 1985
- GMELCH86 Gmelch, B.: Repräsentation multimedialer Dokumente auf der Basis eines Bürodokumentenarchitekturmodells, Diplomarbeit Universität Erlangen-Nürnberg IMMD VI, 1986
- GOLDBERG79 Goldberg, A.; Robson, D.: A Metaphor for User Interface Design, Proc. of the University of Hawaii Systems Science Symposium, Honolulu 1979
- GOLDBERG83 Goldberg, A.; Robson, D.: Smalltalk-80: The Language and its Implementation, Addison-Wesley 1983
- GOODFELLOW86 Goodfellow, M.J.: WHIM, the Window Handler and Input Manager, IEEE Computer Graphics and Applications 6 (1986) 5, S. 46ff
- GORNY84 Gorny, P.; Viereck, A.: Interaktive grafische Datenverarbeitung, Teubner, Stuttgart 1984
- GOULD83 Gould, J.D.; Boies, S.J.: Human Factors Challenges in Creating a Principal Support Office System - The Speech Filing System Approach, ACM Transactions on Office Information Systems 1 (1983) 4, S. 273ff

- GRAY81 Gray, J.: The Transaction Concept: Virtues and Limitations, Proc. of the 7th Conference on Very Large Databases, Cannes 1981
- GUTKNECHT85 Gutknecht, J.: Concepts of the Text Editor Lara, Communications of the ACM 28 (1985) 9, S. 942ff
- GUTTAG77 Gutttag, J.: Abstract Data Types and the Development of Data Structures, Communications of the ACM 20 (1977) 6, S. 396ff
- HACKER78 Hacker, W.: Allgemeine Arbeits- und Ingenieurpsychologie 2. Aufl, Huber, Bern 1978
- HÄRDER83 Härder, T.; Reuter, A.: Concepts for Implementing a Centralized Database Management System, in: Schneider, H.J. (Hrsg.): Proceedings of the International Computing Symposium 1983, Teubner, Stuttgart 1983
- HÄRDER85 Härder, T.; Reuter, A.: Architektur von Datenbanksystemen für Non-Standard-Anwendungen, in: Blaser, A.; Pistor, P. (Hrsg.): Datenbanksysteme für Büro, Technik und Wissenschaft, Springer, Berlin 1985
- HAFER85 Hafer, C.: Untersuchung eines interaktiven Modells zur Dokumentenarchitektur am Beispiel des STAR, Diplomarbeit Universität München, 1985
- HERZOG84 Herzog, O.; Reisig, W.; Valk, R.: Petri-Netze: ein Abriß ihrer Grundlagen und Anwendungen, Informatik-Spektrum 7 (1984) 1, S. 20ff
- HEWITT73 Hewitt, C.: A Universal Modular ACTOR Formalism, Proceoc. of the 3rd IJCAI, Stanford 1973
- HEWITT84 Hewitt, C.; de Jong, P.: Open Systems in: Brodie, M.L. et al. (Hrsg.): On Conceptual Modelling, Springer, New York 1984
- HINES79 Hines, C.; Searle, G.: Automatic Unemployment, Earth Resources Research, London 1979
- HINRICHS83 Hinrichs, K.; Nievergelt, J.: The Grid File: A Data Structure Designed to Support Proximity Queries on Spatial Objects, Bericht Nr. 54, Institut für Informatik, ETH Zürich 1983
- HIRSCHHEIM85 Hirschheim, R.A.: Office Automation - Concepts, Technologies and Issues, Addison-Wesley, Wokingham 1985

- HOFMANN84 Hofmann, P.: Modellierung von Arbeitsabläufen im Büro, Diplomarbeit Universität Erlangen-Nürnberg IMMD VI 1984
- HOFSTADTER80 Hofstadter, D.R.: Gödel, Escher, Bach: An Eternal Golden Braid, Vintage Books, New York 1980
- HOGG83 Hogg, J.; Mazer, M.; Gamvroulas, S.; Tsichritzis, D.: Imail - An Intelligent Mail System, Database Engineering 6 (1983) 3, S. 36ff
- HOGG85a Hogg, J.: Intelligent Message Systems, in: Tsichritzis, D. (Hrsg.): Office Automation, Springer, New York 1985
- HOGG85b Hogg, J.; Nierstrasz, O.M.; Tsichritzis, D.: Office Procedures, in: Tsichritzis, D. (Hrsg.): Office Automation, Springer, New York 1985
- HOPGOOD86 Hopgood, F.R.A. et al. (Hrsg.): Methodology of Window Management, Springer, Berlin 1986
- HORAK84a Horak, W.; Krönert, G.: An Object-Oriented Office Document Architecture Model for Processing and Interchange of Documents, Proc. of the ACM Conference on Office Automation, Toronto 1984
- HORAK84b Horak, W.: Dokument-Architektur und Dokument-Austauschformate - Stand der internationalen Normung, Tagungsband "Offene Multifunktionale Büroarbeitsplätze und Bildschirmtext", Berlin 1984
- HORAK85a Horak, W.; Krönert, G.: Konzept eines Editors basierend auf der Büro-Dokumentenarchitektur ECMA 101, in: Wedekind, H.; Kratzer, K. (Hrsg.): Büroautomation '85, Teubner, Stuttgart 1985
- HORAK85b Horak, W.: Office Document Architecture and Office Document Interchange Formats: Current Status of International Standardization, IEEE Computer 18 (1985) 10, S. 50ff
- IRVING86 Irving, R.H.; Higgins, C.A.; Safayeni, F.R.: Computerized Performance Monitoring Systems: Use & Abuse, Communications of the ACM 29 (1986) 8, S. 794ff
- ISO84a International Standards Organization: DIP 8897, Vocabulary - Part Two of Eighth Working Draft, ISO TC97/SC18/WG8 N8, Genf 1984
- ISO84b International Standards Organization: DIP 8897, Document Markup Metalanguage - Part Six of Eighth Working Draft, ISO TC97/SC18/WG8 N8, Genf 1984

- JONES86 Jones, W.P.; Dumais, S.T.: The Spatial Metaphor for User Interfaces: Experimental Tests of Reference by Location versus Name, ACM Transactions on Office Information Systems 4 (1986) 1, S. 42ff
- KAMBARTEL80 Kambartel, F.: Pragmatische Grundlagen der Semantik, in: Gethmann, C.F. (Hrsg.): Theorie des wissenschaftlichen Argumentierens, Suhrkamp, Frankfurt (Main) 1980
- KARCHER82 Karcher, H.: Büro der Zukunft - Einflußfaktoren der Marktentwicklung für innovative Bürokommunikations-Terminals, Dissertation, Universität München 1982
- KASBERGER86 Kasberger, C.: Verwaltung von Legenden in einer INGRES-Datenbankumgebung, Studienarbeit Universität Erlangen-Nürnberg IMMD VI, 1986
- KAY77a Kay, A.C.; Goldberg, A.: Personal Dynamic Media, IEEE Computer 10 (1977) 3, S. 31ff
- KAY77b Kay, A.C.: Microelectronics and the Personal Computer, Scientific American 237 (1977) 3, S. 230ff
- KAY84 Kay, A.C.: Computer Software, Scientific American 251 (1984) 3, S. 53ff
- KAY85 Kay, A.C.: The User Interface As A Theatrical Mirror, in: Wedekind, H.; Kratzer, K. (Hrsg.): Büroautomation '85, Teubner, Stuttgart 1985
- KIM83 Kim, W.; Lorie, R.; McNabb, D.; Plouffe, W.: Nested Transactions For Engineering Design Databases, IBM Research Report RJ 3934, San Jose 1983
- KOVED84 Koved, L.: The Object Model: A Historical Perspective - Draft, IBM Research Report RC 10809, September 1984
- KOVED86 Koved, L.; Shneiderman, B.: Embedded Menus: Selecting Items in Context, Communications of the ACM 29 (1986) 4, S. 312ff
- KRATZER85 Kratzer, K.: Legends: A Database-Oriented Implementation of Office Procedures, in: Bucco, G.; Valle, G. (Hrsg.): COMPUTING 85 - A Broad Perspective of Current Developments, North-Holland, Amsterdam 1985
- KRATZER86 Kratzer, K.; Schön, D.: Multimediale Datenverarbeitung in der Technischen Dokumentation, in: Clauer, A.; Purgathofer, W. (Hrsg.): AUSTROGRAPHICS '86 - Aktuelle Entwicklungen in der Graphi-

- schen Datenverarbeitung, R. Oldenbourg, Wien München 1986
- KÜHN86 Kühn, P.J.: Dienstintegrierendes Digitalnetz (ISDN), Informatik-Spektrum 9 (1986) 2, S. 132ff
- KÜSPERT86 Küspert, K: Non-Standard-Datenbanksysteme, Informatik-Spektrum 9 (1986) 3, S. 184f
- LAMERSDORF84 Lamersdorf, W.; Müller, G.; Schmidt, J.W.: Language Support for Office Modelling, Proc. of the 10th Int. Conf. on VDLB, Singapore 1984
- LEAVENWORTH78 Leavenworth, B.: An Approach to Office System Simulation, IBM Research Report RC 7133, Yorktown Heights 1978
- LOCHOVSKY83a Lochovsky, F.H.: Improving Office Productivity: A Technology Perspective, in: Tsichritzis, D. (Hrsg.): Beta Gamma, Technical Report CSRG-150, University of Toronto, Toronto, Canada 1983
- LOCHOVSKY83b Lochovsky, F.H.: A Knowledge-Based Approach To Supporting Office Work, Database Engineering 6 (1983) 3, S. 43ff
- LORENZEN78 Lorenzen, P.; Lorenz, K.: Dialogische Logik, Wissenschaftliche Buchgesellschaft, Darmstadt 1978
- LORENZEN79 Lorenzen, P.: Praktische und theoretische Modalitäten, Philosophia Naturalis 17 (1979) 3, S. 261ff
- LYNGBAEK84 Lyngbaek, P.; McLeod, D.: Object Management in Distributed Information Systems, ACM Transactions on Office Information Systems 2 (1984) 2, S. 96ff
- LUM82 Lum, V.Y.; Choy, D.M.; Shu, N.C.: OPAS: An Office Procedure Automation System, IBM Systems Journal 21 (1982) 3, S. 327ff
- MALONE83 Malone, T.W.: How Do People Organize Their Desks? Implications for the Design of Office Information Systems, ACM Transactions on Office Information Systems 1 (1983) 1, S. 99ff
- MARTIN86 Martin, P.; Tsichritzis, D.: Complete Logical Routings in Computer Mail Systems, ACM Transactions on Office Information Systems 4 (1986) 1, S. 64ff
- MAUL85 Maul, G.; Nagl, M.; Schlageter, G.; Zieschang, R.: A Short Note on an Object-oriented and Strongly Typed Programming Language for Data Base Applications, Notizen zu interaktiven

- Systemen o. Jgg. (1985) 14, S. 29ff
- MCDERMOTT76 McDermott, D.: Artificial Intelligence Meets Natural Stupidity, SIGART Newsletter o.Jgg. (1976) 57, S. 4ff
- MERTENS83 Mertens, P.: Decision-Support-Systeme, Informatik-Spektrum 6 (1983) 3, S. 168f
- MERTENS85a Mertens, P.: Zwischenbetriebliche Integration in der EDV, Informatik-Spektrum 8 (1985) 2, S. 81ff
- MERTENS85b Mertens, P.: Forschungsergebnisse zum Nutzen-Kosten-Verhältnis der computergestützten Informationsverarbeitung, in: Ballwieser, W.; Berger, K.-H. (Hrsg.): Information und Wirtschaftlichkeit, Gabler, Wiesbaden 1985
- MERTENS86 Mertens, P.; Allgeyer, K.; Däs, H.: Developments and Applications of Expert Systems in Germany, in: Roos, J.L. (Hrsg.): L'Economique et L'Intelligence Artificielle, Aix-en-Provence 1986
- MEYROWITZ82a Meyrowitz, N.; Van Dam, A.: Interactive Editing Systems: Part I, ACM Computing Surveys 14 (1982) 3, S. 321ff
- MEYROWITZ82b Meyrowitz, N.; Van Dam, A.: Interactive Editing Systems: Part II, ACM Computing Surveys 14 (1982) 3, S. 353ff
- MÖLLER86 Möller, H.; Rosenow, E.: Benutzermodellierung für ein wissensbasiertes MCS-AS-Architekturmodell, Forschungsbericht FB-TA-86-14 WISDOM-Verbundprojekt TRIUMPH-ADLER AG, Nürnberg 1986
- MORBEY85 Morbey, G.: Systematisches Erstellen eines Konzeptionellen Schemas für Dokumente im Büro, Diplomarbeit Universität Erlangen-Nürnberg IMMD VI, 1985
- MORRIS86 Morris, J.H. et al.: Andrew: A Distributed Personal Computing Environment, Communications of the ACM 29 (1986) 3, S. 184ff
- MÜLLER83 Müller, T.; Steinbauer, D.; Wedekind, H.: Control of Versions in Database Applications, Research Report TR 83.09.003 IBM Heidelberg Scientific Center, Heidelberg 1983
- NAFFAH79 Naffah, N.: La Burotique: Definition et justification, Interne Bericht INRIA, Rocquencourt Juni 1979

- NEWMAN80 Newman, W.: Office Models and Office Systems Design, in: Naffah, N.: Integrated Office Systems - Burotics, Proceedings of the INRIA/IFIP TC-6 International Workshop on Integrated Office Systems - Burotics, North-Holland, Amsterdam 1980
- NEWTON85 Newton, S.J.: Graphics & Images In Office Systems, NCC Publications, Manchester 1985
- NICHOLSON85 Nicholson, R.T.: Usage Patterns in an Integrated Voice and Data Communications System, ACM Transactions on Office Information Systems 3 (1985) 3, S. 307ff
- NIEHUIS85 Niehuis, S.; Viktor, F.: Modellierung von Büroprozeduren mit Pr/T-Netzen und Prolog, in: Wedekind, H.; Kratzer, K. (Hrsg.): Büroautomation '85, Teubner, Stuttgart 1985
- NIERSTRASZ85 Nierstrasz, O.M.; Tsichritzis, D.C.: An Object-Oriented Environment for OIS Applications, in: Proc. of the International Conference on Very Large Databases, Stockholm 1985
- NILSSON82 Nilsson, N.J.: Principles of Artificial Intelligence, Springer, Berlin 1982
- ONG84 Ong, J.; Fogg, D.; Stonebraker, M.: Implementation of Data Abstraction in the Relational Database System INGRES, SIGMOD Record 14 (1984) 1, S. 1ff
- ORWELL73 Orwell, G.: Nineteen Eighty-Four, Penguin Books, Basingstoke 1973
- PARKINSON65 Parkinson, C.N.: Parkinson's Law or The Pursuit of Progress, Penguin, Harmondsworth 1965
- PARNAS85 Parnas, D.L.: Software Aspects of Strategic Defense Systems, Communications of the ACM 28 (1985) 12, S. 1326ff
- PANKO82 Panko, R.R.; Sprague, R.: Towards A New Framework For Office Support, Proc. of the 1983 SIGOA Conference, Philadelphia, Juni 1983
- PANKO84 Panko, R.R.: 38 Offices: Analyzing Needs in Individual Offices, ACM Transactions on Office Information Systems 2 (1984) 3, S. 226ff
- RACHE86 Rache, W.; Effelsberg, W.: Die X.400-Empfehlungen für Nachrichtenübermittlungssysteme, Informatik-Spektrum 9 (1986) 1, S. 42f

- RICHARDS82 Richards, J.T.; Boies, S.J.: Speech Filing, in: Endres, A.; Reetz, J. (Hrsg.): Textverarbeitung und Bürosysteme, Oldenbourg, München 1982
- RICHTER83 Richter, G.: Netzmodelle für die Bürokommunikation - Teil 1, Informatik-Spektrum 6 (1983) 4, S. 210ff
- RICHTER84 Richter, G.: Netzmodelle für die Bürokommunikation - Teil 2, Informatik-Spektrum 7 (1984) 1, S. 28ff
- RÖHRLE85 Röhrle, J.: Entwurf und Implementierung eines Relationen-Editors unter INGRES, Studienarbeit Universität Erlangen-Nürnberg IMMD VI, 1985
- RÖHRLE86 Röhrle, J.: Integritätsbedingungen - Konzept und Realisierung mit einem Data-Dictionary, Diplomarbeit Universität Erlangen-Nürnberg IMMD VI, 1986
- ROSENOW86 Rosenow, E.: Adaption der Dialogführung durch Benutzermodelle, Diplomarbeit Universität Erlangen-Nürnberg IMMD VI, 1986
- SAKATA85 Sakata, S.; Ueda, T.: A Distributed Interoffice Mail System, IEEE Computer 18 (1985) 10, S. 106ff
- SCHELLER86 Scheller, A.; Appelt, W.: Derzeitiger Stand der Technik und der internationalen Normungsarbeit auf dem Gebiet der Dokumentenverarbeitung, in: Nake, F. (Hrsg.): Graphik in Dokumenten, Springer, Berlin 1986
- SCHINDLER84 Schindler, S. et. al.: Electronic Office Systems, International Standards/Recommendations/Specifications for Document Exchange and The Committee Support System, Proc. of the 2nd Int. Conference on Introduction of OSI Standards, Ottawa 1984
- SCHOLL85 Scholl, J.: Büroautomation mit CEO, Data General GmbH 1985
- SCHÜRR85 Schürr, A.: Layout von Bürodokumenten im Rahmen eines formalen Modells, Diplomarbeit Technische Universität München, 1985
- SCHUMANN85 Schumann, M.: Computergestützte Eingangspostbearbeitung - Überblick über den Stand und weitere Forschungsaufgaben, in: Wedekind, H.; Kratzer, K. (Hrsg.): Büroautomation '85, Teubner, Stuttgart 1985

- SCHUMANN86 Schumann, M.: Computerunterstützte Eingangspostbearbeitung unter besonderer Berücksichtigung des Expertensystemansatzes, Dissertation Universität Erlangen-Nürnberg, Nürnberg 1986
- SEIGIS86 Seigis, A.: Architektur eines Dokumenten-Editors in einer objektorientierten Büroumgebung, in: Nake, F. (Hrsg.): Graphik in Dokumenten, Springer, Berlin 1986
- SMITH82 Smith, D.C. et al.: Designing the Star User Interface, Byte 7 (1982) 4, S. 242ff
- STAUFER85 Stauffer, H.J.: Piktogramme für Bürocomputer, Forschungsbericht FB-TA-85-6 WISDOM-Verbundprojekt TRIUMPH-ADLER AG, Nürnberg 1986
- STEINBAUER83 Steinbauer, D.: Transaktionen als Grundlage zur Strukturierung und Integritätssicherung in Datenbank-Anwendungssystemen, Arbeitsberichte des IMMD 16 (1983) 13
- STEINBUCH68 Steinbuch, K.: Falsch programmiert, dtv, München 1968
- STONEBRAKER83a Stonebraker, M.; Rubenstein, B.; Guttman, A.: Application of Abstract Data Types and Abstract Indices to CAD Data Bases, Proc. of the Annual SIGMOD Meeting, San Jose 1983
- STONEBRAKER83b Stonebraker, M.; Woodfill, J.; Andersen, E.: Implementation of Rules in Relational Data Base Systems, Database Engineering 6 (1983) 4, S. 65ff
- STONEBRAKER83c Stonebraker, M.; Stettner, H.; Lynn, N.; Kalash, J.; Guttman, A.: Document Processing in a Relational Database System, ACM Transactions on Office Information Systems 1 (1983) 2, S. 143ff
- STONEBRAKER84 Stonebraker, M.; Anderson, E.; Hanson, E.; Rubenstein, B.: QUEL As A Data Type, SIGMOD Record 14 (1984) 2, S. 208ff
- STONEBRAKER85 Stonebraker, M.: Triggers and Inference in Database Systems, Research Report IMAG Grenoble 1985
- STOYAN83 Stoyan, H.; Görz, G.: Was ist objektorientierte Programmierung ?, in: Stoyan, H.; Wedekind, H. (Hrsg.): Objektorientierte Software- und Hardwarearchitekturen, Tagungsband, Teubner, Stuttgart 1983

- STOYAN84 Stoyan, H.: Programming Styles in Artificial Intelligence, in: Laubsch, J. (Hrsg.): GWAI-84, Springer, Berlin 1984
- STRECK85 Streck, G.: Mehrfenstertechnik: Architekturen, Strategien, praktischer Einsatz, Forschungsbericht FB-TA-85-10 WISDOM-Verbundprojekt TRIUMPH-ADLER AG, Nürnberg 1985
- SUCHMAN83 Suchman, L.A.: Office Procedure as Practical Action: Models of Work and System Design, ACM Transactions on Office Information Systems 1 (1983) 4, S. 320ff
- TANENBAUM81 Tanenbaum, A.S.: Computer Networks, Prentice-Hall, Englewood Cliffs NJ 1981
- TESLER81 Tesler, L.G.: The Smalltalk Environment, BYTE 6 (1981) 8, S. 90ff
- THIMBLEBY83 Thimbleby, H.: "What You See Is What You Have Got" - a user engineering principle for manipulative display ?, in: Balzert, H. (Hrsg.): Software Ergonomie, Tagungsband, Teubner, Stuttgart 1983
- TILLINGER86 Tillinger, H.: Bürokommunikation: Standards und Gateways zu öffentlichen Netzen, in: Schulz, A.: Die Zukunft der Informationssysteme - Lehren der 80er Jahre, Springer, Berlin 1986
- TSICHRITZIS80 Tsichritzis, D.: OFS: An Integrated Form Management System, in: Proc. of the 6th International Conference on Very Large Data Bases, Montreal 1980
- TSICHRITZIS82 Tsichritzis, D.: Form Management, Communications of the ACM 25 (1982) 7, S. 453ff
- TSICHRITZIS84 Tsichritzis, D.: Message Addressing Schemes, ACM Transactions on Office Information Systems 2 (1984) 1, S. 58ff
- TSICHRITZIS85a Tsichritzis, D. et al.: A Multimedia Office Filing System, in: Tsichritzis, D. (Hrsg.): Office Automation, Springer, New York 1985
- TSICHRITZIS85b Tsichritzis, D.; Nierstrasz, O.M.: End User Objects, in: Wedekind, H.; Kratzer, K. (Hrsg.): Büroautomation '85, Teubner, Stuttgart 1985
- TSICHRITZIS85c Tsichritzis, D.: Objectworld, in: Tsichritzis, D. (Hrsg.): Office Automation, Springer, New York 1985

- VIERECK86 Viereck, A.: Klassifikation des Mensch-Rechner-Dialogs für eine benutzerfreundliche Dialoggestaltung, Notizen zu interaktiven Systemen o.Jgg. (1986) 15, S. 17ff
- WALBRÜCK86 Walbrück, H.J.: Entwicklung von integrierten Bürokommunikationssystemen - Ansätze und Aktionsparameter, Angewandte Informatik 28 (1986) 8, S. 329ff
- WALTER85 Walter, B.: Multi-Level Synchronization And Nested Transactions In Advanced Information Systems, in: Blaser, A.; Pistor, P. (Hrsg.): Datenbanksysteme für Büro, Technik und Wissenschaft, Springer, Berlin 1985
- WEDEKIND80 Wedekind, H.; Ortner, E.: Systematisches Konstruieren von Datenbankanwendungen, Hanser, München 1980
- WEDEKIND81 Wedekind, H.: Datenbanksysteme I (2. Aufl.), Bibliographisches Institut, Mannheim 1981
- WEDEKIND83 Wedekind, H.: Nullwerte in Datenbanken aus modallogischer Sicht, Softwaretechnik-Trends 3 (1983) 3, S. 44ff
- WEDEKIND84 Wedekind, H.: Supporting the Design of Conceptual Schemata by Database Systems, Proceedings of the IEEE International Conference on Data Processing, Los Angeles 1984
- WEDEKIND85 Wedekind, H.: Büroautomation - eine anthropotechnische Aufgabe, in: Wedekind, H.; Kratzer, K. (Hrsg.): Büroautomation '85, Teubner, Stuttgart 1985
- WEDEKIND86 Wedekind, H.: Technisches Problemlösen und die praktische Modalität der Erreichbarkeit, in: Burrichter, C.; Inhetveen, R.; Kötter, R. (Hrsg.): Technische Rationalität und rationale Heuristik, Ferdinand Schöningh, Paderborn 1986
- WEINBERGER79 Weinberger, C.; Weinberger, O.: Logik, Semantik, Hermeneutik, C.H. Beck, München 1979
- WINSTON84 Winston, P.H.: Artificial Intelligence 2nd ed., Addison-Wesley, Reading MA, 1984
- WOO85 Woo, C.C.; Lochovsky, F.H.: An Object-Based Approach to Modelling Office Work, Database Engineering 8 (1985) 4, S. 14ff

- WOUK72 Wouk, H.: The 'Caine' Mutiny, Fontana Books, Glasgow 1972
- XEROX82 Xerox Corporation: XEROX 8010 Information System (Firmenveröffentlichung 9R80377), Dallas 1982
- YANKELOVICH85 Yankelovich, N.; Meyrowitz, N.; Van Dam, A.: Reading and Writing the Electronic Book, IEEE Computer 18 (1985) 10, S. 15ff
- ZDONIK84 Zdonik, S.B.: Object Management System Concepts, Proc. of the ACM Conference on Office Automation, Toronto 1984
- ZIESCHANG85 Zieschang, R.: Das Bürodatenbanksystem als Basis für die Integration von Text, Daten und Graphik in einem Bürokommunikationssystem, in: Blaser, A.; Pistor, P. (Hrsg.): Datenbanksysteme für Büro, Technik und Wissenschaft, Springer, Berlin 1985
- ZISMAN77 Zisman, M.D.: Representation, Specification, and Automation of Office Procedures, Ph.D. Dissertation University of Pennsylvania, Philadelphia PA 1977
- ZLOOF75 Zloof, M.M.: Query by Example, AFIPS Conf. Proc. 1975 National Computer Conference
- ZLOOF82 Zloof, M.M.: Office-by-Example: A business language that unifies data and word processing and electronic mail, IBM Systems Journal 21 (1982) 3, S. 272ff

Anhang B

Stichwortverzeichnis

- Abhängigkeit
 - konstruktive, 58, 131
- Ablauf, 15
- Abschnitt, 31
- ACID, 191, 193
- Adäquatheit, 170
- Adressierungsschema, 53, 162
- Agent, 189
- Aktenordner, 17
- Aktionsparameter, 13
- Angemessenheit, 8
- Animation, 40
- Annotation, 39
- Anwendungswissen, 171
- Arbeitsausführung, 19
- Arbeitslosigkeit, 21
- Arbeitsplatzrechner, 11
- Atomizität, 191
- Ausnahmebehandlung, 137, 174
- Ausnahmesituation, 175

- Backtracking, 153
- Backward Chaining, 178
- Balkendiagramm, 37
- Beeinflussungsfaktor, 24
- Begründungsdimension
 - horizontale, 95
 - vertikale, 95
- Begründungsrichtung
 - produktive, 96
 - reduktive, 96
- Benutzermodell, 87
 - analytisches, 88
 - empirisches, 88
- Betrachtungskomponente, 42
- Browsing, 42
- Büro, 101
 - basisprogramme
 - siehe Bürobasisprogramme
 - datenbanksystem
 - siehe Bürodatenbanksystem
- Definition
 - organisatorisch, 4
 - technisch, 8
- dokument, 26, 111
- finding, 17
- Geschichte, 6
- grafik
 - siehe Bürografik
- handlung, 138
- informationssystem
 - siehe Büroinformationssystem
- Linien-, 15, 144
- organisation, 13
- principal support, 15
- prozedur
 - siehe Büroprozedur
- reminding, 17
- service-oriented, 15
- Stabs-, 15, 144
- Typ, 15
- Büroautomation, 8, 137, 200
 - Definition, 8
- Bürobasisprogramme, 41
- Bürodatenbanksystem, 2
- Bürografik, 37, 40
- Büroinformationssystem, 9
 - Definition, 9
 - Instanz, 9
- Bürokommunikation, 200
- Büroprozedur, 91, 137
 - Modell
 - prozedurales, 159
 - zielorientiertes, 160
- Büroprozeduren
 - Modelle
 - prozedurale, 161
 - zielorientierte, 168
- Bürosemantik, 171
- Business Graphics

- siehe Bürografik
- CAD, 1
 CAM, 1
 CAQ, 1
 Computer Illiteracy, 7
 CPMCS, 23
 CSD, 73
- Darstellungskomponente, 42
 Datenbanksystem, 2, 204
 Datenendgerät
 -abstraktion, Virtuelle, 58
 Datenfernverarbeitung, 11
 Datentyp, 61
 Abstrakter, 43
 Datentypen, 28
 Bildung von, 28
 Datenverwaltung, multimediale, 57
 Dauerhaftigkeit, 193
 DCA, 73, 201
 Decision-Support-System, 55
 Decision-Support-Systeme, 10, 172
 generische, 56
 Deskriptor, 33
 Dialog, 82, 84, 93
 Anwendungs-, 99
 Gestaltung
 Grundsätze der, 85
 Gestaltungs-, 99
 Kontroll-, 99
 Meta-, 99
 -programmierung, 11
 -schritt, 84
 -struktur, 99
 -transaktion, 11
 Digitale Netzwerke, 11
 DM
 siehe Dokumentmonitor
 Document Markup Metalanguage, 47
 Document Web, 65
 Dokument, 5, 26, 60
 -analyse, 54
 -austausch, 77
 Definition, 27
 Hyper-, 66
 -klasse, 76
 -monitor, 79
 siehe Dokumentmonitor
 Normierung, 73
 Selektion, 33
 Dokumentation, 1
 Dokumentenmodell, 28
 Dokumentmodell, 60
 Dokumentmonitor, 63, 68
 Edierkomponente, 42
 Ein-/Ausgabe
 -abstraktion, Objektorientierte, 58
 Electronic Mail, 51
 Entqualifizierung, 22
 Erfolgsanalyse, 152
 Erreichbarkeit, 145
 Expertensystem, 172
- Farbe, 34
 Fenster, 103
 Kacheltechnik
 siehe Kacheltechnik
 Partitioniertes, 103
 Figure of Merit, 148
 File, 17
 Flächenstruktur, 34
 Flexibilität, 137
 Font, 46
 Formatieren
 inkrementell, 46
 interaktiv, 45
 verzögert, 46
 Formular, 161, 172
 Fotokopiergerät, 10
 Fragment, 61
- GDM, 105
 Architektur, 109
 Katalog, 108, 120
 Geräteabstraktion, Logische, 57
 Geräteunabhängigkeit, 57
 GKS, 34, 75
 Grafik, 33
 Computer-, 33
 -editor, 50
 Externe Repräsentation, 34
 Interne Repräsentation, 35
 Logische Struktur, 33
 Primitive, 35, 50
 Gridfile, 206
 GSML, 73
- Handlung, 5, 139
 Büro-, 138
 elementare, 14
 Interaktions-, 90
 Intersystem-, 142
 Intrasystem-, 142
 Konstitution einer, 14
 materiale, 139
 Nominativ-, 90, 128
 Prädikativ-, 90, 128
 Real-, 5

- regulation, 89
- schema, 140
- Sprech-, 5, 90, 139
- symbolisch, 5
- symbolische, 139
- Träger-, 90, 140
- Vermittlungs-, 140
- Zeige-, 90
- Hierarchisierung, 179
- Hyperdokument, 66
- ICN, 164
- Iemail, 53, 189
- Information
 - akustische, 39
 - audio-visuelle, 39
 - tabellarische, 35
- Information Retrieval, 33
- Informationstransparenz, 20
- Infrastruktur, 20
- Integration, 27
- Integritätsbedingung, 3, 175
- Intelligente Nachricht, 53
- Intention, 139
- Interaktion, 82
- Interscript, 73
- ISO-OSI-Referenzmodell, 1, 209
- Isolation, 192
- Junk Mail, 53
- Kacheltechnik, 106
- Kalenderverwaltung, 8
- Kanal/Instanz-Netz, 166
- Kommunikation, 2, 20
- Kommunikationskanal, 85
- Kommunikationswerkzeuge, 51
- Kompatibilität, 59, 79
 - Ablauf-, 59
 - ER-, 59
 - funktional-reduzierte, 59
 - funktionale, 60
 - repräsentative, 59
 - IR-, 59
- Kompetenzkontinuum, 90
- Konfliktbeseitigung, 138
- Konformität, 79
- Konsistenz, 192
- Kontaktverlust, 22
- Kontext, 173
- Koordinatentransformation, 35
- Kreativität, 14, 148
- Künstliche Intelligenz, 168, 190, 211
- Layout, 32
 - Parameter, 32
- Legende, 177
- Liniendiagramm, 37
- Lokale Netzwerke, 11
- Mail, 51
- Mangelbehebung, 138
- Mark-up, 46
- Maus, 8, 50, 109
- Metakommunikation, 88
- Methode, 181
- Modalität, 145
- Modell
 - kalligrafisches, 34
 - Organisations-
 siehe Organisationsmodell
 - Raster-, 34
 - Vektor-, 34
- MS-DOS, 12
- Nachricht, 180
- Nachrichtenaustausch, 179
- Normalform, Modallogische, 176
- Objekt, 2, 43, 105, 179
 - typ, 105
- ODA, 73
 - Konformitätsstufen, 79
- ODIF, 73, 77
- Office-by-Example, 163
- OFFICETALK, 162, 164
- OFS, 163
- OMEGA, 172
- OPAS, 163
- Operations Research, 10, 56
- Operatoranwendung, 152
- Operatorauswahl, 150
- Organisationsmodell, 57, 80, 111, 130
- Organisationsstruktur, 171, 213
- Paginierung, 32
- Paint-Paradigma, 51
- Performator, 140
- Personal Computer, 11
- Petri-Netz, 164, 166
- Piktogramm, 102
- Pile, 17, 108
- Prädikat/Transitionsnetz, 166
- Präsentation, 82
- Privatsphäre
 - Verlust der, 22
- Problem, 147
 - dialektisches, 148

- interpolativ-lösbares, 148
- synthetisches, 148
- Problemlösen, 138, 145, 147
- Programmiersystem,
 - Objektorientiertes, 12, 179
- Programmierung
 - Dialog-, 11
- Programmierung, Objektorientierte, 179
- Proponent, 95
- Proportionalsschrift, 46
- Protokoll, 181
- Query-by-Example, 163
- Rahmen, 61, 63
- Raster, 34
 - grafik, 34
 - modell, 34
- Rationalisierung, 21
- Repräsentation
 - Externe, 30
 - Interne, 30
- Satz, 31
- Schreibmaschine, 10
- Schreibtischoberfläche, 100
- SCOOP, 166
- Serialisierbarkeitsprinzip, 192
- SGML, 48
- Signatur, 33, 206
- Situationsanalyse, 150
- Smalltalk, 32, 162, 183
- Sprache
 - digitalisierte, 39
- Spreadsheet, 48
- Standard Generalized Markup Language, 48
- Standpunkt, 173, 189
- Stapel, 17, 108
- Struktur
 - Logische, 30
- Subarchitektur, 58
- Suchrichtung, 155
- Tabelle, 35, 48
 - Externe Repräsentation, 37
 - Grafische Ausgabe, 37
 - Interne Repraesentation, 38
 - Logische Struktur, 36
 - Tabellarische Ausgabe, 37
- Tabellenkalkulation, 48
- Tätigkeitsprofil, 15
- Technologie, 8
- Teilhaberbetrieb, 11
- Teilnehmerbetrieb, 11
- Telefon, 10
- Text, 31, 45
 - Externe Repräsentation, 32
 - formatierung, 45
 - Interne Repräsentation, 32
 - Logische Struktur, 31
 - verarbeitung, 45
 - Textverarbeitung, 11
- Thesaurus, 33
- Tortendiagramm, 38
- Transaktion, 3, 138, 190
 - Dialog-, 11
- Traversierungskomponente, 42
- Trennung, 32
- Trigger, 3, 163, 177
- Überbuerokratisierung, 22
- Überlappungstechnik, 106
- Umorientierung, 153
- UNIX, 12
- Vektor, 34
 - grafik, 34
 - modell, 34
- Verarmung
 - geistige, 24
- Version, 189
- Verträglichkeit, 59
- VisiCalc, 49
- Visual Programming, 161
- Werkzeug, 41, 61, 105
 - Bearbeitungs-, 41
 - Edier-, 41
 - GDM-basiertes, 126
 - konventionelles, 126
- Workstation
 - siehe Arbeitsplatzrechner
- Wort, 31, 40
- WYSIWYG, 45, 92
- X.400, 73
- Zeigeinstrument, 35, 50, 89, 109, 114, 128
- Zeile, 32
- Zeitalter
 - der Information, 7
 - industriell, 7
 - vorindustriell, 6
- Zelle, 36
- Zielanalyse, 150

Tabellarischer Lebenslauf

Klaus Kratzer

Diplom-Informatiker

geb. am 13.04.1958

in Nürnberg

1964 - 1968	Besuch der Volksschule Knauerstraße in Nürnberg
1968 - 1972	Besuch des Dürer-Gymnasiums Nürnberg
1972 - 1976	Besuch des Leibniz-Gymnasiums Altdorf bei Nürnberg
06/1976	Erfolgreicher Abschluß der Reifeprüfung
10/1976 - 05/1981	Studium der Informatik an der Friedrich-Alexander-Universität Erlangen-Nürnberg
05/1981	Erfolgreicher Abschluß der Diplomprüfung Informatik
06/1981 - 09/1983	Mitarbeiter bei der TRIUMPH-ADLER AG im Bereich Systemssoftware / Datenbankentwicklung
seit 10/1983	Wissenschaftlicher Mitarbeiter am IMMD (Informatik) VI Lehrstuhl für Datenbanksysteme

In letzter Zeit sind folgende Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung erschienen:

=====

Band 14

=====

- Nr. 1 Pattern Recognition
 (Research at Lehrstuhl für Informatik 5)
 (Februar 1981)

- Nr. 2 Bunke, H.:
 Analyse elektrischer Schaltpläne
 (März 1981)

- Nr. 3 Linster, C.-U.:
 SYMPOS/UNIX - Ein Betriebssystem für homogene
 Polyprozessorsysteme
 (Juni 1981)

- Nr. 4 Akyildiz, J.F., Bolch, G.:
 Analytic Solution Techniques for Queueing Network
 Models of Computer Systems
 (Juli 1981)

- Nr. 5 Lindstedt, W.G.:
 Ein Verfahren zur Erstellung portabler algorithm-
 mischer Laufzeitprogramme
 (Juli 1981)

- Nr. 6 Vollmar, R.:
 Zum Begriff des Parallelismus bei Polyautomaten
 (Juli 1981)

- Nr. 7 Luft, A.L.:
 Rationaler Sprachgebrauch und orthosprachliche
 Standardisierung als Grundlagen zuverlässiger Software-
 Entwicklung und -Dokumentation
 (Januar 1982)

- Nr. 8 Händler, W.:
 PROCEEDINGS OF A WORKSHOP ON TAXONOMY IN COMPUTER
 ARCHITECTURE
 (Februar 1982)

- Nr. 9 Beth, Th., Hess, P., Wirl, K.:
 Materialien zur Kryptographie
 (März 1982)

- Nr. 10 Jahresbericht 1981 der Informatik
 (März 1982)

Band 15
=====

- Nr. 1 Wittmann, A.:
Ein Mechanismus für die Synchronisation paralleler
Prozesse
(Februar 1982)
- Nr. 2 Maehle, E.:
Fehlertolerantes Verhalten in Multiprozessoren -
Untersuchungen zur Diagnose und Rekonfiguration
(März 1982)
- Nr. 3 Bathelt, P.:
Vergleich von Synchronisationsmechanismen
(Mai 1982)
- Nr. 4 Keramidis, S.:
Eine Methode zur Spezifikation und korrekten
Implementierung von asynchronen Systemen
(Juni 1982)
- Nr. 5 Fromm, H.J.:
Multiprozessor-Rechenanlagen:
Programmstrukturen, Maschinenstrukturen und Zuordnungs-
probleme
(Mai 1982)
- Nr. 6 Bley, H.:
Vorverarbeitung und Segmentierung von Stromlaufplänen
unter Verwendung von Bildgraphen
(Juli 1982)
- Nr. 7 Hchl, W.:
Informatik-Sammlung
Katalog mit historischen Erläuterungen
(August 1982)
- Nr. 8 Messerer, M.:
Ein neuer Ansatz zur Parallelisierung von
Compilern
(August 1982)
- Nr. 9 Grosch, J.:
Eine Programmiersprache mit mengentheoretischen
Konstrukten und deren effiziente Implementierung
(August 1982)
- Nr. 10 Kleinöder, W.:
Stochastische Bewertung von Aufgabenstrukturen
für hierarchische Mehrrechnersysteme
(August 1982)

Band 15 Fortsetzung

=====

- Nr. 11 Kneißl, F.:
Realisierung von Makro-Datenflußmechanismen auf
hierarchischen Mehrrechnersystemen
(August 1982)
- Nr. 12 Brendel, W., Fan, Z., Klar, R., Schmielau, W.:
ERES 82
Handbuch und Fallstudie
Handbook and Case Study
(Oktober 1982)
- Nr. 13 Pelz, K.:
Portable Codegenerierung in Übersetzern
für Prozeß-Programmiersprachen
(Oktober 1982)
- Nr. 14 Sommer, U., Steinchen, J.:
Ein Kleinrechner-Grafiksystem zur Dokumentation
von EDV-Abläufen
(Oktober 1982)
- Nr. 15 Hein, H.-W.:
Das Erlanger Spracherkennungssystem
(November 1982)
- Nr. 16 Bolch, G., Bruchner, W.:
Analytische Modelle für symmetrische Mehrprozessor-
anlagen mit dynamischen Prioritäten
(Dezember 1982)
- Nr. 17 Jahresbericht 1982 der Informatik
(März 1983)

Band 16
=====

- Nr. 1 Gall, R.:
Formale Beschreibung des inkrementellen Programmierens-
im-Großen mit Graph-Grammatiken
(März 1983)
- Nr. 2 König, R.:
Beiträge zur algebraischen Theorie der formalen Sprachen
(März 1983)
- Nr. 3 Welbourn, D.B.:
The Design of Mechanical Components and the Development
of DUCT:
17 Years of CAD/CAM in Cambridge University
(April 1983)
- Nr. 4 Woitok, R.:
Die System-Implementierungs-Sprache ALICE
(Mai 1983)
- Nr. 5 Schneider, S., Schwab, P., Renninger, W.:
Wesen, Vergleich und Stand von Software zur Produktion
von Systemen der computergestützten Unternehmensplanung
(Juli 1983)
- Nr. 6 Händler, W., Herzog, U., Hofmann, F., Schneider, H.J.
und Mitarbeiter:
Projektabschlußbericht 1978 - 1982 Erlangen General Purpose
Array
(Juli 1983)
- Nr. 7 Mackert, L.:
Modellierung, Spezifikation und korrekte Realisierung
von asynchronen Systemen
(Juli 1983)
- Nr. 8 Reitenspieß, M.:
Sprachkonstrukte zur Spezifikation und korrekten Implemen-
tierung von Schutzproblemen
(Juli 1983)
- Nr. 9 Weber, K.:
Modellierung von Fehlverhalten mit Berücksichtigung
paralleler Abläufe
(August 1983)
- Nr. 10 Dietsch, H.:
Ein verteiltes Kommunikationssystem für einen Verbund
aus lokalen Mikrorechner-Netzwerken
(Dezember 1983)

Band 16 Fortsetzung

=====

- Nr. 11 Maehle, E., Schmitter, E.:
Workshop Fehlertolerante Mehrprozessor- und Mehr-
rechnersysteme
(Dezember 1983)
- Nr. 12 Volkert, J.:
Multiplikative Abstandsfunktionen und deren Anwendung
in der Bildverarbeitung
(Dezember 1983)
- Nr. 13 Steinbauer, D.:
Transaktionen als Grundlage zur Strukturierung und
Integritätssicherung in Datenbank-Anwendungssystemen
(Dezember 1983)
- Nr. 14 Wilke, P.:
Zusammenhänge und Unterschiede zwischen Graph-Grammatiken
und Petri-Netzen sowie verwandter Systeme
(Dezember 1983)
- Nr. 15 Jahresbericht der Informatik 1983
(März 1984)

Band 17
=====

- Nr. 1 Eberlein, W.:
Architektur technischer Datenbanken für
integrierte Ingenieursysteme
(Februar 1984)
- Nr. 2 Mertens, P., Heigl, M.:
Neuere Entwicklungen der computergestützten
Produktionsplanung, Eignung - Verbindungen -
Entwicklungspfade
(April 1984)
- Nr. 3 Stoyan, H.:
Maschinen- unabhängige Code-Erzeugung als semantik-
erhaltende beweisbare Programmtransformation
(Juli 1984)
- Nr. 4 Brendel, W.:
Funktionaler Entwurf und automatische Synthese
hochintegrierter Schaltkreise
(August 1984)
- Nr. 5 Brietzmann, A.:
Semantische und pragmatische Analyse im Erlanger
Spracherkennungsprojekt
(Oktober 1984)
- Nr. 6 Hoppe, B.:
Deterministisches Zuordnen vielfach durchlaufener
Aufgabensysteme in Multiprozessorsystemen
(Oktober 1984)
- Nr. 7 Fumy, W., Posch, S., Rieß, H.P.:
Materialien zur Codierungstheorie III
(Dezember 1984)
- Nr. 8 Akyildiz, J. F.:
Leistungsanalyse von Multiprozessorsystemen mit
Prozeßkommunikation
(Dezember 1984)
- Nr. 9 Jahresbericht der Informatik 1984
(März 1985)

- Nr. 1 Siegmann, H.:
Wesen, Vergleich und Stand von zehn ausgewählten
kleinrechnerorientierten Planungssprachen
(Februar 1985)
- Nr. 2 Rathke, M.:
Parallelisieren ordnungserhaltender Programmsysteme
für hierarchische Multiprozessorsysteme
(Juni 1985)
- Nr. 3 Geus, L.:
Parallelisierung eines Mehrgitterverfahrens für die
Navier-Stokes-Gleichungen auf EGPA-Systemen
(Juli 1985)
- Nr. 4 Hellmold, K.-U.:
Der Simulationsrechner SIMPLEX
Eine Multiprozessorkonfiguration zur Simulation
von zeitdiskreten Systemen auf GPSS-FORTRAN-Basis
(September 1985)
- Nr. 5 Klebes, G.:
Entwicklung eines botschaftsorientierten Synchroni-
sationsverfahrens für parallele Prozesse und seine
Realisierung im Rahmen eines Realzeitprogrammiersystems
und
Rupprecht, G.:
Ein Konzept zur Integration von allgemeinen Sprach-
konstrukten zur Synchronisation und von gemeinsamen
abstrakten Datentypen mehrerer Prozesse in Modula-2 und
seine Implementierung
- Nr. 6 Schön, D.:
Technisches Informationssystem und Entwurf integrierter
Ingenieurdatenbanken
(Oktober 1985)
- Nr. 7 Vasconcelos, A.J.V. de:
Berechnung nichtstationärer Vorgänge in Drehstromnetzen
mit Raumzeigerkomponenten auf einem Multiprozessor
(Dezember 1985)
- Nr. 8 Baader, F.:
Die S-Varietät $\underline{DS}$ und einige Untervarietäten
(Dezember 1985)
- Nr. 9 Jahresbericht der Informatik 1985
(März 1986)

- Nr. 1 Erhard, W.:
Feldrechner DAP: Invertierung großer Matrizen
(Februar 1986)
- Nr. 2 Fumy, W.:
Über orthogonale Transformationen und fehlerkorrigierende Codes
(März 1986)
- Nr. 3 Ehrlich, U.:
Ein Lexikon für das natürlich-sprachliche Dialogsystem EVAR
(Mai 1986)
- Nr. 4 Mühlfeld, R.:
Verifikation von Worthypothesen
(Juni 1986)
- Nr. 5 Blank, E., Bolch, G.:
Leistungsbewertung fehlertoleranter Systeme
(Juni 1986)
- Nr. 6 Mertens, P., Allgeyer, K., Däs, H., Schumann, M.:
Betriebliche Expertensysteme in deutschsprachigen Ländern
- Versuch einer Bestandsaufnahme -
(Juli 1986)
- Nr. 7 Moritzen, K.:
Softwarehilfsmittel zur Programmierung von Multiprozessoren mit begrenzten Nachbarschaften
Ein Beitrag zur Konfigurationsverwaltung
(September 1986)
- Nr. 8 Kragl, G.:
Eine Programmierumgebung für verteiltes PEARL-Codeerzeugung mittels Programmsynthese
(Oktober 1986)
- Nr. 9 Völck, A.:
Ein vollständiges Mengenkonzept in Programmiersprachen - Realisierung mit unkonventioneller Hardware
(Oktober 1986)

